

KENDALI KECEPATAN PUTARAN MOTOR SMART WHEELCHAIR UNTUK MEREDAM GETARAN PADA JALAN BERGELOMBANG DENGAN METODE PID

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:
Ganda Wibawa Putra
NIM: 145150301111039

**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

KENDALI KECEPATAN PUTARAN MOTOR SMART WHEELCHAIR UNTUK MENEKAN
GETARAN PADA JALAN BERGELOMBANG DENGAN METODE PID

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :

Ganda Wibawa Putra

NIM: 145150301111039

Skripsi ini telah diuji dan dinyatakan lulus pada
27 Desember 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dahnial Syaugy, S.T., M.T., M.Sc.
NIK: 2016078704231002

Wijaya Kurniawan, S.T., M.T.
NIP: 19820125 201504 1 002

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 27 Desember 2018

Ganda Wibawa Putra

NIM: 145150301111038



KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadiran Tuhan Yang Maha Esa, yang telah melimpahkan rahmat, berkah, dan karunia-Nya maka penulis dapat menyelesaikan penelitian beserta penyusunan laporan skripsi dengan judul “Kendali Kecepatan Putaran Motor Smart Wheelchair Untuk Menekan Getaran Pada Jalan Bergelombang Dengan Metode Pid”. Penulis menyadari bahwa selama proses penelitian berlangsung hingga penulisan laporan skripsi ini selesai, terdapat berbagai pihak yang membantu. Oleh karena itu penulis hendak menyampaikan rasa hormat dan ungkapan terimakasih kepada:

1. Kedua orang tua yang sudah memberikan berbagai bentuk dukungan kepada peneliti baik dengan do’a maupun dukungan moral.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika.
3. Bapak Dahnil Syaury, S.T., M.T., M.Sc. dan Bapak Wijaya Kurniawan, S.T., M.T. selaku dosen pembimbing skripsi yang telah meluangkan waktu pikiran dan tenaga untuk membantu membimbing dan mengarahkan peneliti sehingga skripsi ini dapat terselesaikan.
4. Rekan skripsi Devi Ayu Ratnasari, Octavian Metta Wisnu serta rekan-rekan penulis yang telah membantu dan membimbing selama masa perkuliahan hingga pengerjaan skripsi dengan saran maupun motivasi
5. Achmad Fanani Kurniawan yang sama-sama berjuang dalam skripsi tentang *smart wheelchair*.
6. Seluruh civitas akademik di Fakultas Ilmu Komputer yang telah membantu peneliti selama menempuh studi di Universitas Brawijaya Malang

Tidak dapat dipungkiri bahwa dalam pengerjaan penelitian dan penyusunan laporan ini belum sempurna, maka penulis mengharap kritik dan saran yang membangun. Demikian, semoga penelitian ini bermanfaat bagi pembacanya dan pihak yang menggunakannya.

Malang, 16 Desember 2018

Penulis

Gandawibawa007@gmail.com

ABSTRAK

Kursi roda merupakan alat bantu yang digunakan oleh orang yang mengalami kesulitan berjalan menggunakan kaki. Terdapat berbagai macam jenis kursi roda, salah satunya yaitu kursi roda elektrik yang digerakkan oleh motor listrik. Dalam aplikasinya, pengguna kursi roda elektrik di fasilitas umum mengalami ketidak nyamanan. Ketidak nyamanan tersebut berupa getaran yang diterima akibat kondisi jalan yang berbeda-beda serta kontrol kecepatan motor yang statis. Kendali Kecepatan Putaran Motor Smart Wheelchair bekerja dengan cara mengontrol kecepatan motor DC menggunakan metode PID. Masukan sistem didapat dari sensor IMU yang bekerja sebagai pendeteksi kondisi jalan yang tidak rata. Hasil pembacaan sensor IMU berupa nilai 3axis accelerometer yaitu posisi pada sumbu x, sumbu y, dan sumbu z. Masukan dari sensor IMU akan diolah pada NodeMCU menggunakan metode PID untuk menentukan kecepatan motor. Proses PID dimulai dari melakukan pemberian nilai setpoint, lalu pengolahan data dari accelerometer yang telah diubah menjadi vector dan nantinya digunakan sebagai nilai input. Masukan dari sensor akan menentukan nilai getaran untuk diolah menggunakan PID. Setelah nilai output didapat, motor elektrik akan bergerak sesuai dengan besaran nilai output dimana motor dapat menaikkan kecepatan hingga batas tertinggi jika kondisi jalan rata. Sebaliknya, jika kondisi jalan tidak rata maka motor akan menurunkan kecepatan hingga meredam guncangan. Hasil pengujian dari sistem ini berupa sistem dapat meredam getaran mekanis maksimal sebesar 87,9% dan persentase keberhasilan sistem dalam meredam getaran hingga dianggap nyaman berdasarkan standar ISO sebesar 80%

Kata kunci: Kursi Roda Elektrik, NodeMCU, PID, ISO-2631-1

ABSTRACT

A wheelchair is a tool used by people who have difficulty walking using the feet. There are various types of wheel chairs, one of which, namely electric wheelchair which is driven by an electric motor. In its application, the electric wheelchair users in public facilities suffered inconvenience. The inconvenience be accepted due to vibrations of varying road conditions as well as the static motor speed control. Control the speed of rotation of the Motor the Smart Wheelchair works by controlling the speed of a DC motor using PID method. Input system of sensors the IMU working as detector uneven road conditions. The results of the readings of the sensors the IMU in the form value 3axis accelerometer that is the position on the x-axis, y-axis, and z-axis. Input from the sensors the IMU will be processed on the NodeMCU method using the PID to determine the speed of the motor. PID process started off doing the giving the value of the setpoint, and the processing of data from the accelerometer has been transformed into a vector and later used as the input value. The input of the sensor will determine the value of the vibration to be processed using the PID. After the output values obtained, electric motors will move in accordance with the magnitude of value of output where the motor can increase the speed up to the highest limit if road conditions. Conversely, if uneven road conditions then the motor will lower the speed up to dampen the shocks. The results of the testing of this system in the form of percentage of IMU sensors sensitivity of 90% and the percentage of the system's success in absorbing the vibrations of 80%.

Keywords: *Electric Wheelchairs, NodeMCU, PID*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian	2
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori.....	6
2.2.1 PID	6
2.2.2 ISO-2631-1	8
2.2.3 NodeMCU ESP8266.....	9
2.2.4 Sensor IMU.....	10
2.2.5 Kursi Roda elektrik	11
2.2.6 I ² C	12
2.2.7 Arduino IDE	12
BAB 3 METODOLOGI PENELITIAN	14
3.1 Studi Literatur	14
3.2 Rekayasa Kebutuhan	15
3.3 Perancangan Sistem	16
3.4 Implementasi.....	16
3.5 Pengujian dan analisis Sistem.....	17
3.6 Pengambilan Kesimpulan	17

BAB 4 REKAYASA KEBUTUHAN	18
4.1 Deskripsi Umum	18
4.1.1 Perspektif Sistem	18
4.1.2 Ruang Lingkup	18
4.1.3 Karakteristik Pengguna	18
4.1.4 Lingkungan Operasi Sistem	19
4.1.5 Batasan Perancangan dan Implementasi	19
4.1.6 Asumsi dan Ketergantungan	19
4.2 Rekayasa Kebutuhan	19
4.2.1 Kebutuhan Fungsional	19
4.2.2 Kebutuhan Perangkat Keras	20
4.2.3 Kebutuhan Perangkat Lunak	21
BAB 5 PERANCANGAN DAN IMPLEMENTASI	22
5.1 Perancangan Sistem	23
5.1.1 Perancangan Perangkat Keras sensor	23
5.1.2 Perancangan Penempatan sensor	25
5.1.3 Perancangan Perangkat Lunak Pengambilan Data Sensor	25
5.1.4 Perancangan Perangkat Lunak Kontrol PID	27
5.1.5 Perancangan Keluaran motor DC	29
5.2 Implementasi Sistem	29
5.2.1 Implementasi Perangkat Keras	29
5.2.2 Implementasi Penempatan sensor	30
5.2.3 Implementasi Pengambilan Data dari Sensor IMU	31
5.2.4 Implementasi PID	33
5.2.5 Implementasi Keluaran Motor DC	35
BAB 6 PENGUJIAN DAN ANALISIS SISTEM	38
6.1 Pengujian dan analisis Putaran motor DC Pada Jalan Rata dan Pada Jalan Concrete Blok	39
6.1.1 Tujuan	39
6.1.2 Alat	39
6.1.3 Metode	39
6.1.4 Hasil	39
6.2 Pengujian dan Analisis sistem kendali kecepatan untuk mengurangi getaran	41

6.2.1 Tujuan	41
6.2.2 Alat	41
6.2.3 Metode	41
6.2.4 Hasil	41
6.2.5 Analisis	43
6.3 Pengujian dan Analisis sistem kendali kecepatan untuk mengurangi getaran dengan standar ISO-2631-1	43
6.3.1 Tujuan	43
6.3.2 Alat	43
6.3.3 Metode	43
6.3.4 Hasil	43
6.3.5 Analisis	45
BAB 7 PENUTUP	46
7.1 Kesimpulan	46
7.2 Saran	46
Daftar Pustaka	47
LAMPIRAN	48



DAFTAR TABEL

Tabel 2.1 Tabel Konversi Parameter Tuning Ziegler-Nichols	8
Tabel 2.2 Level kenyamanan tubuh pengguna ketika terpapar getaran mekanis..	9
Tabel 2.3 Spesifikasi Node MCU.....	10
Tabel 2.4 Spesifikasi sensor IMU.....	11
Tabel 4.1 Kebutuhan fungsional perangkat keras	21
Tabel 4.2 Kebutuhan fungsional perangkat lunak	21
Tabel 5.1 Koneksi pin Sensor GY-521 dan nodeMCU	24
Tabel 5.2 Konversi Parameter Tuning Ziegler-Nichols.....	29
Tabel 5.3 perancangan keluaran sistem motor dc.....	29
Tabel 5.4 Kode program pengambilan data sensor IMU	31
Tabel 5.5 Tabel Konversi Parameter Tuning Ziegler-Nichols.....	34
Tabel 5.6 Kode program kontrol PID.....	34
Tabel 5.7 Kode program keluaran motor DC.....	35
Tabel 6.1 Tabel Lokasi Percobaan.....	38
Tabel 6.2 level kenyamanan tubuh pengguna ketika terpapar getaran mekanis berdasarkan ISO 2631-1 (1997)	44
Tabel 6.3 Hasil Pengujian kontrol PID dengan berbagai kondisi jalan.....	44

DAFTAR GAMBAR

Gambar 2.1 Parameter Pendekatan Ziegler-Nichole	8
Gambar 2.2 Kurva respon quarter amplitude decay	8
Gambar 2.3 Node MCU ESP8266 V1 Iolin	9
Gambar 2.4 Sensor IMU	10
Gambar 2.5 Kursi Roda Elektrik	11
Gambar 2.6 Aliran Data I ² C	12
Gambar 2.7 Logo Software Arduino IDE	13
Gambar 3.1 Diagram Alir Alur Penelitian	14
Gambar 3.2 Blok diagram desain sistem	16
Gambar 5.1 Skematik diagram alir kinerja sistem secara keseluruhan	22
Gambar 5. 2 Diagram Blok Sistem	23
Gambar 5.3 Koneksi pin Sensor GY-521 dan nodeMCU	24
Gambar 5.4 Lokasi peletakkan sensor dan nodeMCU	25
Gambar 5.5 Register power management	26
Gambar 5.6 Register Accelerometer configuration	27
Gambar 5.7 Diagram alir proses PID	28
Gambar 5.8 kurva respon dari metode quarter decay	29
Gambar 5.9 Peletakan kontroller	30
Gambar 5.10 Peletakan sistem	30
Gambar 5.11 Penempatan sensor	31
Gambar 5.12 Respon sistem ketika nilai Ku sebesar 5	34
Gambar 6. 1 Contoh salah satu lokasi yang digunakan pada pengetesan	38
Gambar 6.2 Nilai Pwm Jalan Rata	40

Gambar 6.3 Nilai pwm jalan tidak rata	40
Gambar 6.4 Hasil pembacaan sensor pada sistem tanpa PID	42
Gambar 6.5 Hasil setelah menggunakan control kecepatan dengan PID.....	42
Gambar 6.6 Hasil Pengujian Perbandingan Getaran	42
Gambar 6.7 Hasil nilai getaran	45



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Salah satu alat bantu di dunia medis yang dipergunakan untuk orang yang mengalami kesulitan berjalan dengan kaki, yang umumnya disebabkan oleh cacat, penyakit, ataupun cedera adalah kursi roda. Alat ini dapat digerakkan sendiri dengan menggunakan tangan, digerakkan dengan bantuan pihak lain, ataupun menggunakan mesin otomatis. Ada banyak jenis kursi roda yang bisa digunakan, mulai dari kursi roda untuk anak-anak, kursi roda manual, hingga kursi roda elektrik. Kursi roda yang paling umum digunakan untuk semua kegiatan adalah kursi roda manual. Kursi roda manual mampu digerakkan dengan memanfaatkan tangan penggunanya sendiri ataupun dengan bantuan orang lain. Sedangkan kursi roda untuk anak-anak memiliki ukuran yang lebih kecil dibanding kursi roda pada umumnya namun kegunaan serta fungsinya tidaklah jauh berbeda dengan kursi roda manual. Kursi roda listrik, ialah kursi roda yang berpengerak motor listrik, biasanya jenis kursi roda ini dipergunakan oleh penderita cacat atau penderita cacat ganda yang tidak sanggup untuk menjalankan kursi roda sendiri. Untuk dapat menjalankan kursi roda mereka, bisa dilakukan dengan mengontrol tuas seperti *joystick* untuk menggerakkan maju, mengubah arah jalan kursi roda untuk berbelok ke kiri ataupun berbelok ke kanan dan untuk menghentikan jalannya kursi roda. Rata-rata kursi roda listrik memiliki alat untuk mengisi ulang daya, alat tersebut dapat dihubungkan dengan stop kontak (Rosidawati, et al., 2008).

Dalam beberapa tahun terakhir, pengembangan fasilitas yang menunjang manula untuk beraktivitas menjadi poin utama dalam pembangunan dan juga sebuah masalah sosial. Orang tua yang menggunakan kursi roda di fasilitas umum pun mengalami kesulitan dan ketidak nyamanan. Semenjak penggunaan dan peralatan medis terus meningkat, kenyamanan dalam penggunaan kursi roda di berbagai kondisi akan menjadi fokus utama. Dalam studi ini, berdasarkan getaran yang diserap atau yang diterima akibat kondisi jalan ketika berjalan menggunakan kursi roda elektrik di kondisi jalan yang berbeda-beda. Disini dibutuhkan penelitian hubungan getaran yang terjadi pada kursi roda elektrik dengan torsi motor elektrik serta kecepatan motor elektrik saat digunakan di kondisi jalan yang berbeda (Wang, et al., 2017).

Dengan adanya kemajuan teknologi dibidang teknologi medis, serta meningkatnya penggunaan peralatan tersebut, maka kenyamanan serta efektivitas dari alat tersebut menjadi tantangan di masa depan, saat ini penggunaan kursi roda elektrik sedang marak digunakan. Kenyamanan dan keamanannya pun menjadi poin utama penelitian, terutama di jalan yang tidak rata. Keadaan jalan yang tidak tentu serta kontrol akan kecepatan motor akan menimbulkan ketidak nyamanan bagi pengguna kursi roda elektrik karena efek

getaran yang didapatkan. Kombinasi dari jalan yang buruk serta kontrol akan torsi motor yang besar juga dapat mengakibatkan kecelakaan karena pengguna dapat terlempar dari kursi roda elektriknya. Maka dari itu pentingnya mengatur kecepatan motor selain agar pengguna kursi roda elektrik merasa nyaman pengguna juga aman dari kondisi jalan yang bergelombang (Noriaki Hirose, 2012).

Dari hal tersebut dapat diketahui kemajuan teknologi dibidang medis pun sangat berpengaruh bagi kehidupan sosial, semakin banyaknya kebutuhan kursi roda elektrik yang aman serta nyaman cepat atau lambat akan sangat dibutuhkan. Maka dari itu penulis mengusulkan solusi dengan melakukan penelitian yang berjudul "Kendali Kecepatan Putaran Motor Smart Wheelchair Untuk Meredam Getaran Pada Jalan Bergelombang Dengan Metode PID". Karena, perlu diketahui pentingnya untuk mengatur kecepatan motor di kondisi jalan yang tidak rata, karena selain memberikan keamanan, mengatur kecepatan motor pun bisa menjadi solusi murah untuk mengurangi guncangan yang terjadi di kondisi jalan yang tidak rata, daripada menggunakan suspensi magnetic yang relative mahal dan sulit dicari namun tetap memenuhi standar ISO-2631 mengenai '*Mechanical vibration and shock – Evaluation of human exposure to whole-body vibration*'. Penggunaan PID sendiri dikarenakan komputasinya jauh lebih ringan dan cocok untuk digunakan pada mikrokontroler dibanding menggunakan metode kontrol lainnya yang memiliki komputasi jauh lebih berat

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat dirumuskan masalah seperti berikut:

1. Bagaimana performa sistem menghadapi kondisi jalan bergelombang?
2. Bagaimanakah cara mengimplementasikan PID untuk kecepatan dan torsi motor dari hasil deteksi jalan bergelombang?
3. Apakah sistem mampu lolos kualifikasi ISO-2631?

1.3 Tujuan Penelitian

Berdasarkan Rumusan Masalah yang telah diuraikan di atas, maka tujuan dari penelitian sebagai berikut:

1. Mengetahui peforma sistem menghadapi kondisi jalan bergelombang.
2. Sistem dapat menghasilkan kecepatan motor serta torsi motor berdasarkan hasil proses PID .
3. Sistem mampu lolos kualifikasi ISO-2631

1.4 Manfaat Penelitian

Manfaat penelitian ini secara praktis adalah:

1. *Difabel*, dapat memberikan kenyamanan bagi difabel yang menggunakan kursi roda

2. Orang Tua, dapat membuat mereka lebih tenang akan kenyamanan pengguna kursi roda
3. Masyarakat, dapat meyakinkan mereka bahwa kursi roda elektrik juga mampu memberikan kenyamanan tanpa harus dibantu orang lain selayaknya kursi roda pada umumnya
4. Secara teoritis, hasil penelitian ini dapat bermanfaat kepada peneliti lain yang ingin meneliti lebih lanjut terhadap smart wheel chair baik dari segi kenyamanan maupun keamanan.

1.5 Batasan Masalah

Batasan-batasan dari permasalahan di atas meliputi:

1. Sistem diujikan di jalan seputaran daerah universitas brawijaya yang menggunakan *concrete block*
2. Sistem diujikan hanya menggunakan 3 parameter (gerak sensor pada sumbu x dan y, gerak sensor pada sumbu z,)
3. Sistem memberikan perintah untuk mengurangi torsi pada motor listrik agar getaran yang diterima berkurang

1.6 Sistematika Pembahasan

Sistematika pembahasan dari Kendali Kecepatan Putaran Motor Smart Wheelchair Untuk Menekan Getaran Pada Jalan Bergelombang Dengan Metode PID adalah sebagai berikut:

Bab 1 Pendahuluan

Pada bab ini akan dijelaskan mengenai latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, dan batasan masalah dan sistematika pembahasan

Bab 2 Landasan Kepustakaan

Pada bab ini akan dijelaskan mengenai studi literatur, dasar teori dari penelitian Kendali Kecepatan Putaran Motor Smart Wheelchair Untuk Menekan Getaran Pada Jalan Bergelombang Dengan Metode PID

Bab 3 Metodologi

Pada bab ini akan dijelaskan mengenai metode dan langkah-langkah yang dilakukan dalam pengerjaan penelitian ini. Dimulai dari studi literatur, analisis kebutuhan, desain sistem, implementasi, pengujian dan analisis, dan pengambilan kesimpulan dari penelitian Kendali Kecepatan Putaran Motor Smart Wheelchair

Bab 4 Rekayasa Kebutuhan

Bab ini membuat kebutuhan yang harus dipenuhi guna keberhasilan berjalannya sistem sesuai dengan tujuan dibuatnya sistem. Rincian dari bab ini berupa deskripsi secara umum rekayasa kebutuhan, baik kebutuhan fungsional sistem dan kebutuhan non fungsional sistem

Bab 5 Perancangan dan Implementasi

Bab ini membahas perancangan sistem baik secara hardware dan software sebelum kemudian diimplementasikan. Bab ini juga membahas implementasi sistem secara bertahap, berupa perancangan dan implementasi hardware beserta perancangan dan implementasi metode yang digunakan

Bab 6 Pengujian dan Analisis Sistem

Bab pengujian dan analisis berisi pengujian secara keseluruhan sistem dan analisis yang dilakukan terhadap sistem. Pengujian dan analisis dilakukan pada hardware

Bab 7 Penutup

Bab penutup membahas mengenai kesimpulan dan saran yang dihasilkan dari keseluruhan proses pengerjaan penelitian yang telah dilakukan



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Dalam penulisan skripsi, peneliti juga menggali informasi mengenai penelitian-penelitian sebelumnya yang relevan sebagai bahan referensi.

1. Jurnal Tianqi Wang, Jun'ichi Kaneko, Kazuyuki Kojima Universitas Saitama yang berjudul "Study on Relevance Between Electric Wheelchair Riding Comfort and User Exposure to Whole-Body Vibration" penelitian terkait dengan kursi roda elektrik dengan tujuan mengurangi getaran yang diserap dari jalanan yang buruk dengan mengatur kecepatan dari motor. Pada jurnal tersebut, smartwheel chair yang digunakan adalah Yamaha JW yang dipadukan dengan sensor gerak buatan Bosch yaitu BNO055 9 DOF. Penelitian ini dites di 4 jenis jalan, yaitu linoleum, asphalt, braile block dan pintu masuk lift. (Wang, et al., 2017)
2. Jurnal Noriaki Hirose, Kazutoshi Sukigara and Yuji Tsusaka dari Toyota Central R&D Labs., INC Suppression of Roll Vibration for Personal Mobility Robot using Driving Torque of Wheels. Penelitian terkait pada kemampuan torsi motor untuk menangani roll vibration, yang dapat mengganggu kenyamanan bahkan menyebabkan pengguna terjatuh yang diaplikasikan pada prototype Personal Mobility Robot milik Toyota yang saat ini sedang dikembangkan. (Noriaki Hirose, 2012)
3. Jurnal Afdy Clinton, Dahnial Syauqy, Fitri Utaminingrum dari Teknik Informatika Universitas Brawijaya yang berjudul "sistem monitoring rpm roda smart wheelchair pada halaman web berbasis ajax menggunakan sensor optocoupler" yang membahas tentang pendeteksian kecepatan roda dengan sensor optocoupler yang dipasang pada roda belakang kursi roda, sehingga mampu membaca kecepatan putaran yang dihasilkan oleh motor listrik. (Clinton, et al., 2018)

Perbedaan dari penelitian yang dilakukan sebelumnya dilakukan oleh Tianqi Wang adalah terletak pada sensor yang digunakan. Disini penulis menggunakan sensor GY-521, sedangkan pada penelitian sebelumnya menggunakan sensor milik Bosch. Perbedaan yang terdapat pada penelitian yang dilakukan Noriaki Hirose adalah terletak pada apa yang dikontrol, jika pada penelitian sebelumnya yang di kontrol adalah torsi dari motor, sedangkan penulis mengatur kecepatan dari motor tersebut. Selanjutnya pada penelitian yang dilakukan oleh Afdy Clinton hal yang di gunakan oleh penulis adalah penggunaan mikrokontroller NodeMCU. Hal ini diperlukan karena 2 penelitian sebelumnya menggunakan mikrokontroller yang dibuat sendiri.

2.2 Dasar Teori

Pada Sub-bab dasar teori berisi penjelasan tentang teori-teori yang digunakan dalam penelitian ini diantaranya adalah PID, Sensor IMU, dan Mikrokontroler node MCU.

2.2.1 PID

Kontroler PID (dari singkatan bahasa Inggris: *Proportional–Integral–Derivative controller*) merupakan kontroler dengan mekanisme umpan balik yang biasanya dipakai pada sistem kontrol pada dunia industri. Sebuah kontroler PID secara kontinyu menghitung nilai kesalahan sebagai beda antara setpoint yang diinginkan dan variabel proses yang terukur. Kontroler mencoba untuk meminimalkan nilai kesalahan setiap waktu dengan penyetelan *variabel kontrol*, seperti posisi keran kontrol, damper, atau daya pada elemen pemanas, ke nilai baru yang ditentukan oleh jumlahan:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.1)$$

Keterangan :

$mv(t)$ = output dari pengontrol PID atau Manipulated Variable

K_p = konstanta Proporsional

T_i = konstanta Integral

T_d = konstanta Derivatif

$e(t)$ = error (selisih antara set point dengan level aktual)

Berdasarkan persamaan (2.1) dimana nilai K_p , K_i dan K_d , semuanya positif, menandakan koefisien untuk term proporsional, integral, dan derivatif, secara berurutan (atau P , I , dan D). Pada model ini,

- P bertanggung jawab untuk nilai kesalahan saat ini. Contohnya, jika nilai kesalahan besar dan positif, maka keluaran kontrol juga besar dan positif.
- I bertanggung jawab untuk nilai kesalahan sebelumnya. Contoh, jika keluaran saat ini kurang besar, maka kesalahan akan terakumulasi terus menerus, dan kontroler akan merespon dengan keluaran lebih tinggi.
- D bertanggung jawab untuk kemungkinan nilai kesalahan mendatang, berdasarkan pada rate perubahan tiap waktu.

Karena kontroler PID hanya mengandalkan variabel proses yang terukur, bukan pengetahuan mengenai prosesnya, maka dapat secara luas digunakan. Dengan penyesuaian (*tuning*) ketiga parameter model, kontroler PID dapat memenuhi kebutuhan proses. Respon kontroler dapat dijelaskan dengan bagaimana responnya terhadap kesalahan, besarnya overshoot dari setpoint, dan derajat osilasi sistem. Penggunaan algoritme PID tidak menjamin kontrol optimum sistem atau bahkan kestabilannya.

Beberapa aplikasi mungkin hanya menggunakan satu atau dua term untuk memberikan kontrol sistem yang sesuai. Hal ini dapat dicapai dengan mengontrol parameter yang lain menjadi nol. Kontroler PID dapat menjadi kontroler PI, PD, P atau I tergantung aksi apa yang digunakan. Kontroler PI biasanya adalah kontroler paling umum.

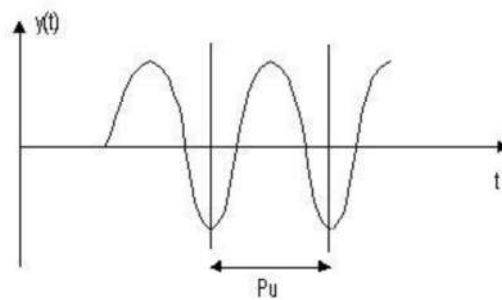
Term proporsional menghasilkan nilai keluaran yang berbanding lurus dengan nilai kesalahan. Responnya dapat diatur dengan mengalikan kesalahan (error) dengan konstanta K_p , disebut konstanta *gain* proporsional atau *gain* kontroler. Gain yang besar menghasilkan perubahan yang besar pada keluaran untuk suatu nilai kesalahan tertentu. Namun, jika gain terlalu besar, sistem dapat menjadi tidak stabil. Sebaliknya, gain yang bernilai kecil maka respon keluaran juga kecil, sehingga kontroler menjadi kurang responsif/sensitif, akibatnya tindakan kontrol menjadi terlalu kecil bila ada gangguan.

Peranan dari term integral berbanding lurus dengan besar dan lamanya error. Integral dalam kontroler PID adalah jumlahan error setiap waktu dan mengakumulasi offset yang sebelumnya telah dikoreksi. Error terakumulasi dikalikan dengan *gain* integral K_i dan menjadi keluaran kontroler. Term integral mempercepat perpindahan proses menuju setpoint dan menghilangkan *steady-state error* yang muncul pada kontroler proporsional. Namun, karena integral merespon terhadap error terakumulasi dari sebelumnya, maka dapat menyebabkan overshoot.

Term derivative atau turunan error pada proses dihitung dengan menentukan kemiringan error setiap waktu dan mengalikan perubahan tiap waktu dengan *gain* derivatif K_d . Aksi derivatif memprediksi perilaku sistem dan kemudian memperbaiki waktu tinggal dan stabilitas sistem. Aksi derivatif jarang digunakan pada industri diperkirakan hanya 25% kontroler karena akibatnya pada stabilitas sistem pada aplikasi dunia nyata. (Dan Chen, 2002).

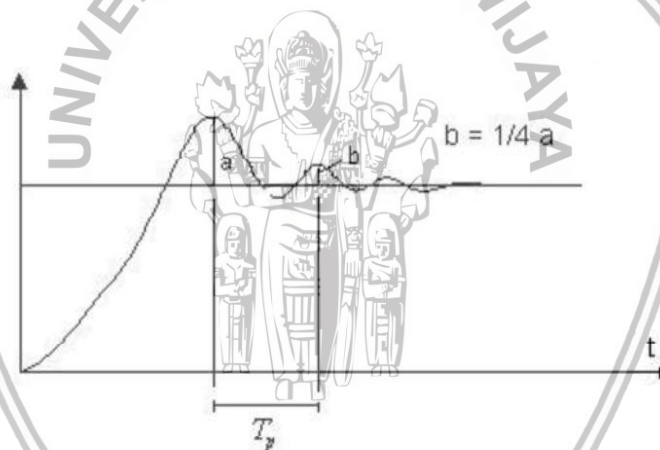
Ketika kontroler *PID* diterapkan pada sebuah sistem hal yang sangat penting untuk dilakukan adalah melakukan tuning untuk mendapatkan respon sistem yang diinginkan. Tuning *PID* dilakukan untuk menemukan nilai K_p , K_i , K_d yang bisa menghasilkan respon sistem yang paling bagus. Ada beberapa teknik dalam melakukan tuning *PID* ini, dan yang paling banyak digunakan adalah melalui pendekatan *Ziegler Nichols* yang ditemukan oleh J.G Ziegler dan N.B Nichols.

Untuk mendapat nilai K_p , K_i , K_d yang harus dilakukan diawal adalah dengan melakukan penentuan parameter K , T_d , T_i seperti pada Gambar 2.1 dengan cara menarik garis gradien pada grafik respon sistem. Setelah didapatkan nilai parameternya baru dihitung K_p , K_i , K_d sesuai Tabel 2.1.



Gambar 2.1 Parameter Pendekatan Ziegler-Nichole
Sumber: (Ogata, 2010)

Karena tidak semua proses dapat memiliki keadaan osilasi dengan amplitude tetap, maka dapat digunakan metode *quarter amplitude decay*. Pada metode ini, sistem dibuat hingga menghasilkan respon berbentuk *quarter amplitude decay* dengan parameter seperti pada Gambar 2.2. Sedangkan perhitungan parameter kontroler PID sama dengan yang digunakan pada metode *Ziegler-Nichols* seperti pada Tabel 2.1



Gambar 2.2 Kurva respon quarter amplitude decay
Sumber: (Gunterus, 1994)

Tabel 2.1 Tabel Konversi Parameter Tuning Ziegler-Nichols

Tipe Kontroler	K_p	T_i	T_d
P	$0.5 \cdot Ku$	∞	0
PI	$0.45 \cdot Ku$	$0.5 \cdot Pu$	0
PID	$0.6 \cdot Ku$	$0.5 \cdot Pu$	$0.125 \cdot Pu$

Sumber: (Gunterus, 1994)

2.2.2 ISO-2631-1

ISO-2631-1 merupakan aturan tentang nilai ambang batas getaran yang dibuat untuk menjaga seseorang dari resiko yang dihasilkan oleh getaran mekanis. Getaran mekanis dapat didefinisikan sebaga getaran yang ditimbulkan alat

mekanis, dari getaran yang dihasilkan tersebut hingga menuju ke tubuh pengguna sehingga menimbulkan akibat akibat yang tidak diinginkan yang mempengaruhi konsentrasi, kelelahan gangguan kesehatan mata dan syaraf dan lain sebagainya. Aturan ini merupakan aturan yang diakui secara internasional dan dibuat oleh Organisasi Standar International yang berpusat di jenewa. Peraturan ISO mengenai getaran mekanis lebih menekankan pada kenyamanan dan keamanan pengguna suatu alat. Pengguna alat yang terpapar getaran dari suatu alat melampaui nilai ambang batas yang telah ditentukan secara kontinyu akan merasakan kelelahan sehingga mempengaruhi produktivitas pengguna. Detail pengelompokan getaran sesuai dengan ISO seperti pada Tabel 2.2.

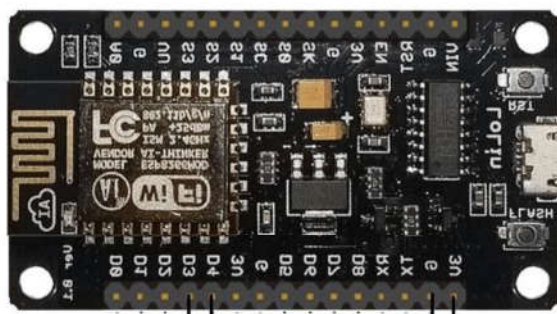
Tabel 2.2 Level kenyamanan tubuh pengguna ketika terpapar getaran mekanis

Nilai percepatan (m/s ²)	Skala kenyamanan
Kurang dari 0.315	Nyaman
0.315 – 0,5	Sedikit kurang nyaman
0,5 – 0,8	Agak tidak nyaman
0,8 – 1,25	Tidak nyaman
1,25 – 2	Sangat tidak nyaman
Lebih dari 2	Amat sangat tidak nyaman

Sumber: (International Organization for Standardization, 1997)

2.2.3 NodeMCU ESP8266

NodeMCU merupakan sebuah platform open source dan development kit yang support IoT. NodeMCU menggunakan perangkat keras berupa Sistem On Chip ESP8266 serta dapat diprogram dengan bahasa pemrograman Lua yang memiliki kesamaan logika dan penyusunan pemrograman dengan bahasa c namun berbeda syntax atau dapat menggunakan sketch Arduino IDE untuk membantu dalam membuat program prototype yang akan dikerjakan. Board yang didasari pada Sistem On Chip ESP8266, yang mengintegrasikan GPIO, PWM (pulse Width Modulation), I2C, dan Analog to Diigital Converter semua dalam satu board seperti Gambar 2.2 berikut.



Gambar 2.3 Node MCU ESP8266 V1 Iolin

Sumber: (Zerynth, 2016)

Berikut adalah spesifikasi Node MCU:

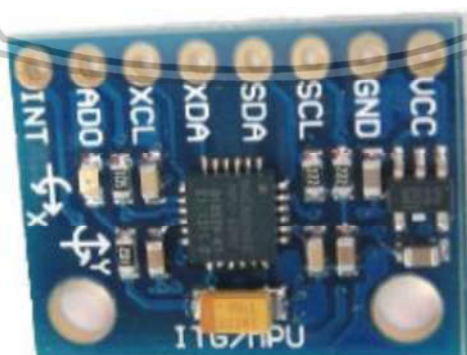
Tabel 2.3 Spesifikasi Node MCU

Spesifikasi Node MCU ESP8266 V3 Iolin	
Mikrokontroler	ESP8266
Catu Daya	3.3v-5v
Ukuran Board	57mm x 30 mm
GPIO	13 Pin
10 bit ADC pin	1pin
USB to Serial converter	CH340G
WiFi	802.11 b/g/n
Frekuensi	2.4GHz
USB Port	Micro USB
Flash Memory	4MB
SRAM	64KB
Kecepatan Clock	80 MHz

Sumber: (Zerynth, 2016)

2.2.4 Sensor IMU

Merupakan sebuah modul sensor Inertial Measurement Unit (IMU) seperti pada Gambar 2.4 yang menggunakan chip MPU-6050 dengan 3 axis accelerometer dan 3 axis gyroscope. MPU6050 sudah memiliki Digital Motion Processors yang digunakan untuk mengolah data mentah dari masing-masing sensor. Sensor ini menggunakan protokol I2C untuk berkomunikasi dengan mikrokontroler yang dilengkapi dengan FIFO buffer dan built-in sinyal interup. Data dari sensor dapat diletakkan pada FIFO buffer terlebih dahulu sebelum dikirim, kemudian interrupt signal akan memberi informasi kepada mikrokontroler setelah data siap dibaca. Spesifikasi lebih lengkap dapat dilihat di tabel 2.4.



Gambar 2.4 Sensor IMU

Sumber: (Arduino, 2018)

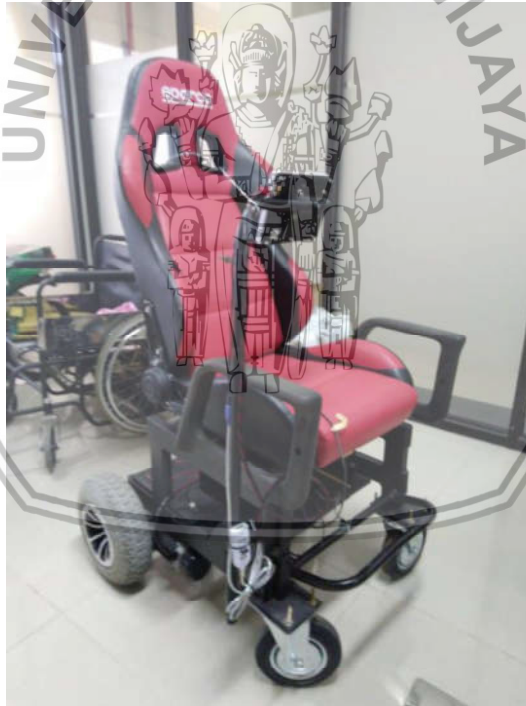
Tabel 2.4 Spesifikasi sensor IMU

Spesifikasi sensor IMU MPU-6050	
Chip	MPU 6050
Catu Daya	3v-5v
Gyroscope range	+250, 500, 1000, 2000/s
Acceleration range	2,4,8,16 g
Mode komunikasi	I2c
Ukuran	20.3mmx15.6mm

Sumber: (Arduino, 2018)

2.2.5 Kursi Roda elektrik

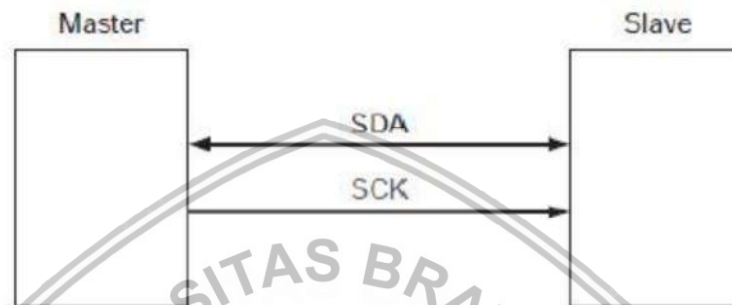
Merupakan salah satu alat bantu terutama bagi penyandang cacat kaki untuk dapat berpindah dari suatu tempat ke tempat lain baik di dalam ruangan maupun diluar ruangan, tempat datar maupun jalan menanjak atau menurun kursi roda yang digerakkan dengan actuator berupa motor listrik yang bisa digunakan untuk perjalanan yang cukup jauh bagi difabel yang tidak mampu menggerakkan sendiri kursi roda (Hilberink, 2006).

**Gambar 2.5 Kursi Roda Elektrik**

Untuk menggerakkan kursi roda elektrik cukup dengan menggunakan sebuah tuas seperti *joystick* untuk menjalankan maju, mengubah arah kursi roda, belok ataupun mengerem. Bentuk dari kursi roda elektrik dapat dilihat pada Gambar 2.4 .

2.2.6 I²C

I²C (inter-integrated circuit) merupakan suatu koneksi yang menyediakan komunikasi antar perangkat agar terintegrasi satu sama lain seperti rtc dan sensor. Komunikasi pada i2c memiliki sifat synchronous, menggunakan protocol dan hanya menggunakan 2 kabel untuk komunikasi, yaitu SCL (synchronous clock) dan SDA (synchronous data), hal inilah yang membuat berbeda dengan komunikasi SPI. Data dikirimkan dari master ke slave secara berurutan, kemudian setelah komunikasi tersebut selesai maka data dari slave menuju master bisa dikirim.



Gambar 2.6 Aliran Data I²C

Seperti pada gambar 2.5, pada perangkat i2c diperluakn 2 buah pin open-drain 2 arah serta pull-up resistor untuk setiap pin yang nantinya akan berlaku sebagai gerbang logika and. Master merupakan perangkat yang menentukan atau menghasilkan clock untuk sistem, menginisiasi, serta untuk memutuskan transmisi. Slave merupakan suatu perangkat yang dialamatkan oleh master dan menerima clock. Baik slave maupun master dapat mentransmisikan data maupun menerima (Barnett, 2007).

I2c merupakan sebuah protocol komunikasi serial yang dimana setiap bit data dikirim melalui jalur SDA yang disinkronisasikan dengan pulsa clock yang ada pada jalur SCL. Arah data yg melewati SDA tidak dapat berubah saat jalur clock ada pada saat kondisi high. (Mazidi, 2011)

Dalam penggunaanya, setiap data atau alamat yang ditransmisikan harus dibentuk kedalam sebuah paket data sepanjang 9bit dengan konfigurasi 8bit di kirim melalui jalur SDA oleh transmitter dan bit ke 9 berfungsi sebagai acknowledge/not acknowledge oleh receiver.

2.2.7 Arduino IDE

Arduino IDE (Intergrated Development Environment) dengan logo seperti pada Gambar 2.6 merupakan suatu lingkungan terintegrasi yang berfungsi untuk melakukan perkembangan. Disebut sebagai lingkungan karena pada software inilah user membuat fungsi-fungsi yang akan ditanamkan perangkat perangkat arduino maupun perangkat-perangkat yang kompatibel dengan Arduino IDE.



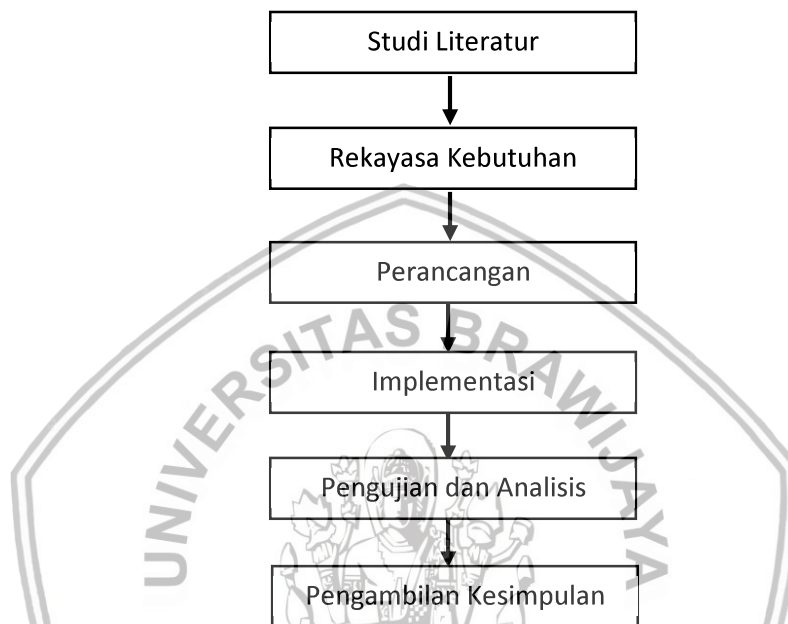
Gambar 2.7 Logo Software Arduino IDE

Sumber: (Arduino, 2018)

Arduino IDE menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman ini sudah dilakukan perubahan dengan tujuan untuk mempermudah pemula dalam melakukan pemrograman. Arduino IDE dibuat berdasarkan bahasa JAVA dan merupakan pengembangan dari software processing dan telah dirombak sedemikian rupa khusus untuk pemrograman dengan perangkat arduino maupun perangkat yang kompatibel dengan Arduino IDE. Arduino IDE juga telah dilengkapi dengan library C/C++ yang disebut dengan Wiring sehingga membuat operasi input dan output menjadi lebih mudah.

BAB 3 METODOLOGI PENELITIAN

Pada bagian ini dijelaskan tentang metode dan langkah-langkah yang dilakukan dalam pengerjaan penelitian ini. Penelitian bersifat implementatif. Berikut merupakan tahapan-tahapan metodologi penelitian yang diambil, tergambar pada diagram alir yang ditunjukkan Gambar 3.1.



Gambar 3.1 Diagram Alir Alur Penelitian

Berdasarkan Gambar 3.1, dapat diuraikan langkah-langkah metodologi penelitian yang akan dilakukan. Pertama akan dipelajari literatur-literatur yang berhubungan dengan penelitian ini. Lalu akan dilakukan analisis kebutuhan sistem yang meliputi kebutuhan fungsional dan non-fungsional. Berikutnya akan didesain sistem untuk menerapkan PID pada kecepatan motor berdasarkan data dari sensor IMU. Kemudian Implementasi sistem dengan menggunakan NodeMCU. Selanjutnya pengujian sistem dilakukan di jalan bergelombang untuk men-tes apakah sistem berjalan dengan baik atau tidak. Jika sudah diuji maka hasil dari pengujian tersebut akan dianalisis tingkat keberhasilannya, jika berhasil maka akan berlanjut ke penarikan kesimpulan.

3.1 Studi Literatur

Pada penelitian ini, dilakukan studi literatur untuk menjadi landasan dari desain dan implementasi sistem ini. Studi literatur mengenai PID, penggunaan sensor IMU untuk mendeteksi getaran, beserta bagaimana cara menimplementasikannya ke bahasa C++ ke NodeMCU menggunakan Arduino IDE. Studi literatur juga tentang bagaimana caranya agar PID dapat menghasilkan kenyamanan dalam menggunakan kursi roda berdasarkan standar ISO.

3.2 Rekayasa Kebutuhan

Rekayasa kebutuhan diperlukan untuk mengidentifikasi dan menganalisis apa-apa saja yang dibutuhkan agar dapat memenuhi kebutuhan penelitian ini. Kebutuhan analisis dibedakan dalam kebutuhan fungsional dan non-fungsional.

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah suatu kebutuhan tentang apa saja yang dapat dilakukan sistem, informasi apa yang didapat dari sistem, dan bagaimana hasil akhir dari sistem. Kebutuhan fungsional dari sistem sebagai berikut:

1. Sistem mendapatkan *input* berupa nilai guncangan pada sumbu x,y,z. Sistem mendapatkan *input* berupa perubahan nilai *accelerometer* pada sumbu x,y,z yang terjadi pada saat melewati jalan tidak rata
2. Sistem memproses hasil *input* untuk mengatur kecepatan motor agar kursi roda yang digunakan berkurang guncangannya dengan PID. Dari hasil *input* yang didapatkan diproses menggunakan PID untuk mengurangi nilai pwm maksimum kecepatan motor
3. Sistem dapat mengeluarkan suatu *output* tergantung dari besaran guncangan yang diterima oleh kursi roda. Dari pengurangan nilai pwm kecepatan motor akan melambat sehingga tumbukan antar kondisi jalan yang tidak rata dengan roda depan kursi roda

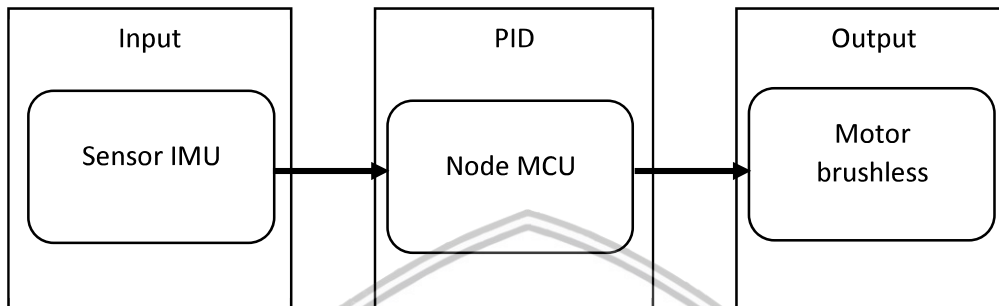
3.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan terhadap perangkat keras dan perangkat lunak yang digunakan untuk penelitian ini. Kebutuhan non-fungsional dari sistem adalah sebagai berikut:

1. Perangkat keras :
 - a. Node MCU
 - b. Motor brushless
 - c. Sensor IMU
2. Perangkat Lunak :
 - a. Arduino IDE
 - b. Library PID

3.3 Perancangan Sistem

Rancangan sistem memuat langkah-langkah perencanaan suatu sistem agar sistem dapat berjalan sesuai dengan analisis kebutuhan. Untuk memenuhi kebutuhan sistem yaitu menghasilkan sistem yang dapat mengklasifikasi dan menentukan suatu tindakan berdasarkan tahapan level, langkah-langkahnya dapat dilihat dari Gambar 3.2.



Gambar 3.2 Blok diagram desain sistem

Berdasarkan Gambar 3.2 terdapat 3 tahapan pada perancangan sistem yang dapat dijelaskan sebagai berikut:

1. Berdasarkan nilai dari 3axis accelerometer yang berasal dari sensor IMU akan menjadi *input* sistem
2. Parameter tersebut menjadi *input* NodeMCU dimana di dalam NodeMCU terjadi penentuan kecepatan berdasarkan PID
3. Keluaran sistem akan ditentukan berdasarkan *input* dimana jika terdapat guncangan yang besar maka kecepatan motor brushless pun melambat

3.4 Implementasi

Implementasi berisikan beberapa langkah yang diambil agar sistem dapat terwujud sesuai dengan desain sistem atau perancangan sistem. Berdasarkan perancangan sistem, implementasi dapat meliputi hal-hal berikut:

1. Penerapan *hardware* sistem dimana *hardware* ini terdiri dari 1 Sensor IMU, 2 motor brushless dan 1 Mikrokontroler NodeMCU. Sensor IMU menjadi *hardware* masukan untuk NodeMCU, lalu motor brushless akan menjadi output, dimana output kecepatan putaran motor dc akan ditentukan tergantung dari masukan
2. Penerapan pengambilan data dari sensor IMU
3. Penerapan kontrol PID untuk mengatur kecepatan motor brushless berdasarkan masukan yang didapat dari sensor IMU
4. Penerapan hasil keluaran pada motor brushless

3.5 Pengujian dan analisis Sistem

Pengujian dan analisis sistem dilakukan untuk menguji tingkat keberhasilan sistem dalam menentukan kecepatan kursi roda untuk *user* berdasarkan masukan yang didapat dari sensor imu. Pengujian pertama dilakukan dengan membandingkan hasil sebelum menggunakan PID dengan sesudah menggunakan PID untuk mengetahui apakah sistem memberikan hasil yang signifikan. Pengujian kedua dilakukan dengan standar ISO-2631-1 untuk mengetahui apakah sistem telah berfungsi dan serta sesuai standar ISO.

3.6 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan berdasarkan analisis dan pengujian terhadap sistem yang sudah berhasil ditemukan set point PID dan menghasilkan luaran yang sesuai. Berdasarkan hasil pengujian, maka dapat ditentukan luaran yang tepat yang akan dibutuhkan untuk kenyamanan *user* sesuai dengan masukan. Jika luaran yang diharapkan belum sesuai maka akan dilakukan kembali tahapan implementasi dan meninjau ulang keadaan sistem baik keadaan *hardware* maupun *software*.



BAB 4 REKAYASA KEBUTUHAN

4.1 Deskripsi Umum

Bab ini membahas secara menyeluruh mengenai rekayasa kebutuhan yang diperlukan pada Sistem kontrol kecepatan putaran motor *smart wheelchair* untuk menekan getaran pada jalan bergelombang dengan metode PID untuk dilanjutkan kedalam tahap implementasi sistem. Metode PID disini berfungsi untuk mengatur kecepatan kursi roda agar energi dari hantaman jalan dengan roda kursi roda dapat berkurang, sehingga getaran yang dirasakan pun berkurang. Rekayasa kebutuhan diperlukan agar sistem mampu bekerja sesuai dengan tujuan pembuatan sistem

4.1.1 Perspektif Sistem

Sistem kontrol kecepatan putaran motor *smart wheelchair* mengatur kecepatan putaran motor kursi roda di jalan yang tidak rata dengan perhitungan PID. Sistem akan mengolah input berupa nilai *g-force* dari getaran mekanis yang didapat dari accelerometer pada sensor IMU, kemudian sistem akan menentukan kecepatan motor yang akan dihasilkan. Sistem ini menggunakan accelerometer pada sensor IMU sebagai input, nodeMCU sebagai mikrokontroler pemroses input, dan motor akan mengeluarkan kecepatan sesuai hasil perhitungan

4.1.2 Ruang Lingkup

Ruang lingkup sistem kontrol kecepatan putaran motor *smart wheelchair* adalah penggunaan NodeMCU sebagai mikrokontroler, metode PID sebagai pengontrol kecepatan, hasil keluaran pengolahan data berupa kecepatan motor dc. Rincian dari ruang lingkup adalah sebagai berikut:

1. Ruang Lingkup Sensor

Ruang lingkup sensor pada sistem ini berupa sebuah accelerometer pada sensor mpu 6050. Sensor tersebut akan mendeteksi *g-force* dari getaran mekanis yang dihasilkan dari jalan yang tidak rata sehingga menyebabkan guncangan pada pengguna kursi roda. Sensor ini akan beroperasi pada tegangan 5v.

2. Ruang Lingkup Output

Ruang lingkup output pada sistem ini berupa kecepatan putaran motor dc yang akan melambat ketika menemui jalan yang tidak rata sehingga mengurangi getaran mekanis yang diterima pengguna kursi roda

4.1.3 Karakteristik Pengguna

Sistem ini merupakan purwarupa serta pengujian dilakukan oleh penulis. Pengguna dalam sistem ini adalah pengguna tunggal karena sistem masih berupa prototype dengan output berupa kecepatan putaran motor dc.

4.1.4 Lingkungan Operasi Sistem

Lingkungan operasi sistem yang mendukung berjalannya sistem ini adalah jalan paving blok, dan ruangan dengan lantai keramik. Pengambilan data pengujian akan dilakukan dilingkungan fakultas ilmu komputer Universitas Brawijaya.

4.1.5 Batasan Perancangan dan Implementasi

Berikut ini adalah perancangan dan implementasi sistem:

1. Accelerometer pada sensor IMU bertindak mendeteksi kondisi jalan yang tidak rata
2. Node MCU bertindak sebagai mikrokontroler yang digunakan untuk memproses PID serta mengambil data sensor
3. Sistem dapat mendeteksi kondisi jalan yang bergelombang
4. Keluaran dari sistem berupa kecepatan motor yang diatur otomatis

4.1.6 Asumsi dan Ketergantungan

Dalam penerapan sistem terdapat asumsi dan ketergantungan, berikut ini adalah asumsi dan ketergantungan dari sistem:

1. Diasumsikan jalanan yang dilewati lantai keramik diruangan, jalanan paving.
2. Daya pada kursi roda menggunakan aki sedangkan daya pada nodemcu menggunakan laptop

4.2 Rekayasa Kebutuhan

Rekayasa kebutuhan menjelaskan mengenai kebutuhan dari sistem secara keseluruhan dan mendetail supaya sistem dapat beroperasi dengan baik dan sesuai dengan perancangan yang telah dibuat. Bagian ini membahas kebutuhan fungsional dan kebutuhan non fungsional sistem yang akan dibuat.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional yaitu kebutuhan yang bersifat wajib dipenuhi agar tujuan dibuatnya sistem ini dapat terwujud. Kebutuhan tersebut mengenai sesuatu yang dapat dilakukan oleh sistem ataupun hasil apa yang didapat dari sistem. Berikut adalah kebutuhan fungsional dari sistem:

1. Sistem dapat mendeteksi jalan tidak rata
 - a. Penjelasan dan Prioritas
Input didapat dari *g-force* yang dihasilkan dari getaran mekanis kursi roda yang kemudian dideteksi oleh accelerometer pada sensor IMU. Fungsi ini memiliki prioritas tinggi karena akan menjadi sumber data input untuk diproses pada mikrokontroler.

b. Respon Sistem

Ketika roda depan kursi roda melewati jalan tidak rata, sensor accelero akan mendeteksi getaran mekanis berupa *g-force* yang dihasilkan dari benturan yang menandakan bahwa jalan bergelombang, nilai kemiringan tersebut dikirim ke nodeMCU untuk diproses

c. Data Uji

Data uji sistem kontrol kecepatan putaran motor smart wheelchair berupa accelerometer pada sensor imu yang ditempatkan dekat dengan roda kursi roda yang melewati jalan tidak rata untuk mendapatkan nilai *gforce* akibat getaran mekanis pada posisi sumbu x, posisi sumbu y dan posisi sumbu z. Data uji diperlukan untuk pengujian mendeteksi kondisi jalan yang tidak rata.

2. Sistem melakukan pemrosesan kontrol dengan PID di nodeMCU

a. Penjelasan dan Prioritas

Sistem melakukan pemrosesan kontrol PID untuk mengurangi getaran yang diterima dengan cara mengurangi kecepatan motor dc berdasarkan nilai sensor. Pada saat berada dalam kondisi jalan yang rata maka kecepatan motor akan berada pada posisi maksimal, pada saat kondisi seperti ini lah akan ditentukan sebagai kondisi set point untuk PID.

b. Respon Sistem

Sistem ini mendapat masukan data kondisi jalan dari accelerometer pada sensor IMU. Kemudian data tersebut diproses menggunakan PID untuk menentukan kecepatan motor.

3. Sistem menghasilkan kecepatan motor sesuai dengan kondisi jalan yang dilewati

a. Penjelasan dan Prioritas

Fungsi ini digunakan sebagai *actuator*, setelah melewati tahap kontrol PID. Hasil perhitungan menggunakan metode PID menghasilkan kecepatan motor yang sesuai dengan kondisi jalan yang dilalui

b. Respon Sistem

Setelah sistem memproses data input menggunakan metode PID maka sistem akan menggerakkan motor sesuai kecepatan yang dibutuhkan

4.2.2 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras sistem kontrol kecepatan motor berupa MPU 6050, Node MCU, dan motor DC. Kebutuhan perangkat keras sistem ini dan fungsinya akan dijelaskan secara rinci pada table 4.1 berikut.

Tabel 4.1 Kebutuhan fungsional perangkat keras

Kebutuhan perangkat keras	fungsi
MPU 6050	Sensor input untuk mendeteksi kemiringan jalan
Node MCU	Mikrokontroler yang akan memproses input yang didapat dengan PID menjadi sebuah output
Motor dc	Output dari proses yang dilakukan di NodeMCU berupa kontrol kecepatan

4.2.3 Kebutuhan Perangkat Lunak

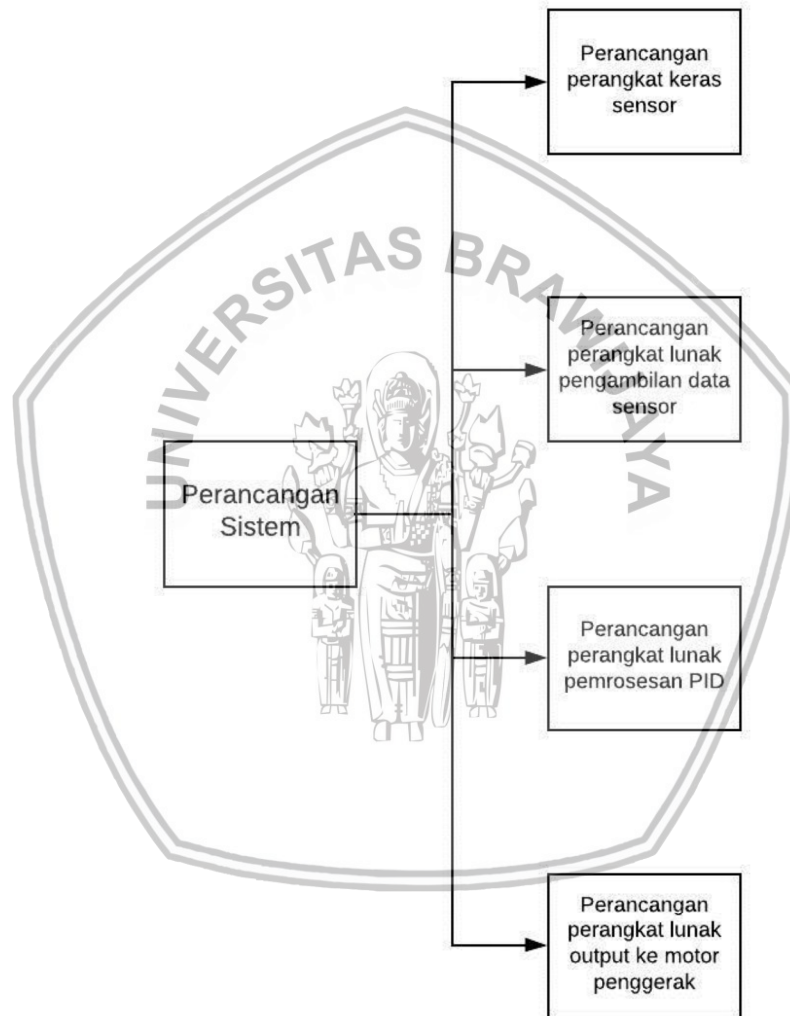
Kebutuhan perangkat lunak sistem ini digunakan untuk memprogram nodeMCU serta melakukan implementasi PID. Dalam implementasi PID dilakukan menggunakan library PID, hal ini dilakukan karena pada library PID telah dilakukan beberapa penyempurnaan dijelaskan pada tabel 4.2 berikut.

Tabel 4.2 Kebutuhan fungsional perangkat lunak

Kebutuhan perangkat lunak	fungsi
Arduino IDE	Penulisan serta upload program arduino
Library PID	Menghitung nilai PID dengan lebih cepat serta akurat

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dibahas tentang Perancangan dan Implementasi dari sistem kontrol kecepatan putaran motor smart wheelchair yang akan dibangun, diantaranya adalah perancangan sistem perangkat keras secara keseluruhan dan perancangan sistem perangkat lunak secara serta implementasi sistem perangkat keras secara keseluruhan, dan implementasi perangkat lunak.

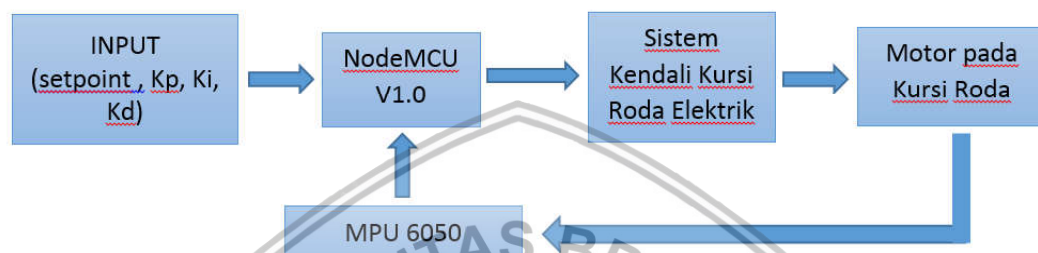


Gambar 5.1 Skematik diagram alir kinerja sistem secara keseluruhan

Berdasarkan Gambar 5.1 perancangan Sistem Kendali Kecepatan Putaran Motor Smart Wheelchair Untuk Menekan Getaran Pada Jalan Bergelombang Dengan Metode PID dibagi menjadi perancangan perangkat keras sensor, perancangan perangkat lunak pengambilan data sensor, perancangan perangkat lunak kontrol PID, dan perancangan perangkat lunak keluaran motor penggerak.

5.1 Perancangan Sistem

Pada sub bab ini akan dijabarkan mengenai tahapan perancangan yang dimulai dari perancangan *hardware*, perancangan *software*, lalu perancangan *prototype*. Pada perancangan *hardware* meliputi sensor MPU 6050 dengan kontroller unit, serta peletakkan pada komponen sistem. Sedangkan pada perangkat lunak terdiri dari pengambilan data dari sensor MPU 6050, sistem utama dan kendali PID serta pergerakan motor. Diagram blok dari sistem yang akan dibuat dapat dilihat pada Gambar 5.2.

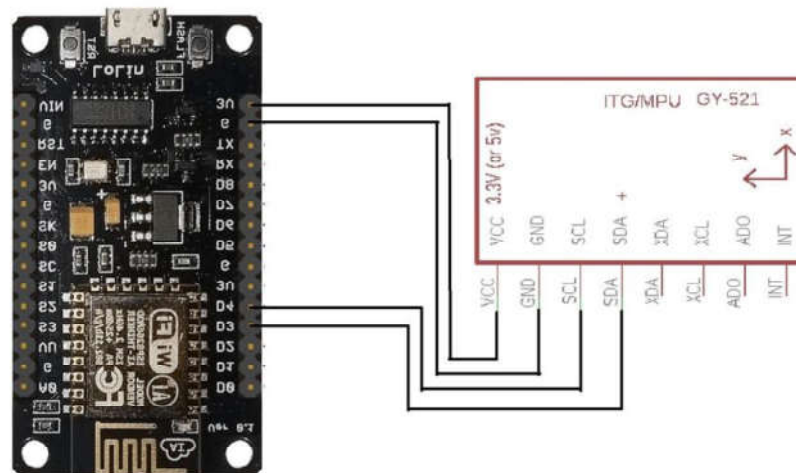


Gambar 5. 2 Diagram Blok Sistem

Pada diagram blok diatas dapat dilihat bahwa sistem diawali dengan melakukan penentuan *setpoint*, nilai Kp, Ki, Kd kemudian data tersebut diolah pada NodeMCU yang kemudian menghasilkan *output* dari sistem berupa nilai besaran PWM yang menghasilkan kecepatan yang berbeda. Dari hasil output tersebut menghasilkan getaran yang akan dibaca oleh sensor MPU 6050. Dari nilai sensor tersebut di kirim ke NodeMCU untuk di cek apakah output sudah sesuai dengan apa yang diinginkan.

5.1.1 Perancangan Perangkat Keras sensor

Salah satu yang utama dari sistem ini adalah sensor inersia, pada sub bab ini akan dijelaskan perancangan sensor inersia tersebut. Sensor inersia yang tersedia dipasaran umumnya adalah MPU6050 6 *degree of freedom* yang terdiri dari *accelerometer* dan *gyroscope* dalam satu *board*. Sensor tersebut menjadi satu dengan komponen-komponen penunjang lainnya dalam sebuah *breakout board* GY-521. Gambar 5.3 akan menggambarkan koneksi antara sensor dengan kontroller unit nodeMCU milik kursi roda.



Gambar 5.3 Koneksi pin Sensor GY-521 dan nodeMCU

Sensor IMU GY-521 berada pada sisi sebelah kanan dari gambar, terdapat 4 buah pin yang digunakan yaitu:

Tabel 5.1 Koneksi pin Sensor GY-521 dan nodeMCU

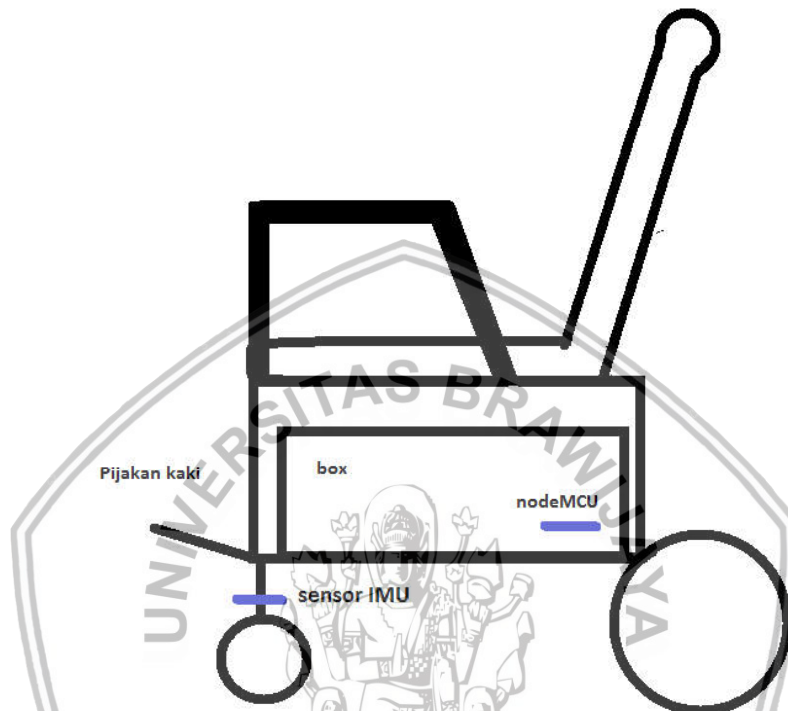
Sensor GY-521	Kontroler unit pin
VCC	Pin 3V
GND	GND
SDA	D3
SCL	D4

Berdasarkan Tabel 5.1 untuk sensor ini bisa digunakan hingga 2 buah dengan memanfaatkan pin ADO dari sensor yang digunakan sebagai VCC untuk merubah alamat sensor tersebut. Pada gambar 5.2 pin ADO dibiarkan tidak terkoneksi karena pin tersebut telah diberi *internal pull down* resistor di dalam *breakout board* GY-521 sehingga menjadikannya bernilai 0 atau LOW. Pada Tabel 5.1 merupakan keterangan koneksi pin antara Sensor GY-521 dengan node MCU. Sensor disambungkan dengan sumber daya dari pin vcc 3v nodeMCU. Pin SDA disambungkan pada pin D3 dan pin SCL dan disambungkan pada pin D4 node MCU.

Node MCU selaku mikrokontroler pemroses PID pada sistem ini dihubungkan dengan sensor IMU untuk mendapatkan masukan berupa nilai *accelerometer* pada sumbu x, sumbu y dan sumbu z. Nilai yang didapatkan akan digunakan untuk menentukan apakah jalan dalam keadaan rata atau bergelombang. Jika jalan dalam keadaan bergelombang, maka PID akan menghitung nilai kesalahan sebagai beda antara setpoint yang diinginkan dan variabel proses terukur. Keluaran sistem yang didapat dari pemrosesan PID berupa percepatan atau perlambatan motor yang mengontrol jalannya kursi roda.

5.1.2 Perancangan Penempatan sensor

Posisi sensor IMU akan diletakkan pada bagian rangka antar roda kiri dan roda kanan. Penempatan sensor pada bagian depan dikarenakan pada posisi itu getaran diterima terlebih dahulu. Untuk lebih jelasnya peletakkan sensor dapat dilihat pada Gambar 5.4.



Gambar 5.4 Lokasi peletakkan sensor dan nodeMCU

Alasan mengapa sensor diletakkan didekat roda adalah karena pada posisi tersebut sensor dapat mendeteksi getaran dengan lebih baik dibandingkan diletakkan didalam box. Serta sesuai dengan syarat standar ISO-2631-1 dalam peletakkan sensor yang harus berada dekat dengan sumber getaran dan titik tumpu kaki pengguna.

5.1.3 Perancangan Perangkat Lunak Pengambilan Data Sensor

Konfigurasi sensor dilakukan pada saat pertama kali sensor tersebut diaktifkan, ada berbagai hal yang perlu diatur agar sensor dapat bekerja seperti yang diinginkan. Diantaranya adalah menonaktifkan *sleep mode* pada sensor, mengatur *full scale range* dan *sensitivity scale factor* dari *Accelerometer*.

1. Register power management 1

Menurut pemetaan register dari sensor MPU6050 untuk menonaktifkan *sleep mode* maka harus mengakses register *PWR_MGMT_1* atau bisa disebut sebagai register *power management 1* yang beralamat di 0x6b kemudian mengganti nilai pada bit ke 6 yang merupakan bit untuk mengatur *sleep mode*, Gambar 5.5 berikut merupakan gambaran isi dari register power management 1.

4.28 Register 107 – Power Management 1 PWR_MGMT_1

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6B	107	DEVICE_RESET	SLEEP	CYCLE	-	TEMP_DIS	CLKSEL[2:0]		

Gambar 5.5 Register power management

Pada register *power management 1*, ada hal lain yang perlu diatur selain untuk mengatur *sleep mode*, diantaranya adalah

a. CLKSEL[2:0]

MPU 6050 telah menyediakan *internal oscillator* sebesar 8MHz. Pada bagian bit 0 hingga bit 2 digunakan untuk memilih sumber *clock sensor*. Maka dari itu, *clock sensor* perlu diaktifkan dengan cara menuliskan nilai 0 (nol) pada ketiga bit CLKSEL tersebut

b. CYCLE

Cycle merupakan bit yang berfungsi untuk memanggil sensor dalam rangka mengambil data Accelerometer dalam beberapa waktu tertentu selama sensor dalam kondisi *sleep mode*. Dikarenakan sensor sudah tidak lagi dalam mode *sleep mode*, maka diperlukan untuk menonaktifkan fungsi maka diperlukan untuk menonaktifkan fungsi CYCLE ini dengan menulis nilai 0 pada bit 6.

c. DEVICE_RESET

DEVICE_RESET merupakan bit yang berguna untuk mengatur ulang ulang sensor kembali ke pengaturan pabrik dan terletak pada bit 7. Karena sensor tidak akan direset, maka bit ini akan diatur dengan nilai 0 (nol).

Seluruh bit pada register power management telah ditentukan nilainya dari bit 0 hingga bit 7 berisi 0 (nol), dari hal tersebut nilai yang harus dituliskan pada register power management 1 dalam biner adalah 0b0000000 yang jika di ubah ke hexa menjadi 0x00

2. Register ACCEL_CONFIG

Register ACCEL_CONFIG merupakan register yang digunakan untuk mengkonfigurasi *accelerometer* untuk memilih *Full Scale Range* yang akan digunakan sesuai dengan ruang lingkup kerja sistem. *Full Scale Range* dibagi menjadi 4 pilihan rentang bacaan mualin dari AFS_SEL 0 sampai 3 dengan rentang bacaan maksimal:

- AFS_SEL 0 untuk $\pm 2\text{ g}$ dan *Sensitivity Scale Factor* sebesar 16.384 LSB/g
- AFS_SEL 1 untuk $\pm 4\text{ g}$ dan *Sensitivity Scale Factor* sebesar 8.192 LSB/g
- AFS_SEL 2 untuk $\pm 8\text{ g}$ dan *Sensitivity Scale Factor* sebesar 4.096 LSB/g
- AFS_SEL 3 untuk $\pm 16\text{ g}$ dan *Sensitivity Scale Factor* sebesar 2.048 LSB/g

AFS_SEL 0 merupakan yang paling sensitif dimana nilai yang dihasilkan dari AFS_SEL 0 adalah 16.384 untuk $\pm 2\text{ g}$, dengan begitu semakin besar hasil yang

didapat maka semakin banyak juga pergerakan yang dapat dideteksi, namun pergerakan yang mampu dideteksi tidak boleh melewati tekanan $\pm 2 g$.

Sistem yang akan dibangun membutuhkan sensitifitas yang tinggi untuk mampu mendeteksi getaran mekanis yang terjadi serta tekanan pergerakan rendah dibawah $\pm 2 g$ maka diputuskan menggunakan *AFS_SEL 0*. Untuk menggunakan *AFS_SEL 0* pada konfigurasi Accelerometer dapat dilakukan dengan cara menulis keseluruhan bit pada register 0X1C dengan 0 (nol) sesuai dengan Gambar 5.6 yang menggambarkan register konfigurasi accelerometer dan cara memilih *AFS_SEL 0*

4.5 Register 28 – Accelerometer Configuration ACCEL_CONFIG

Type: Read/Write

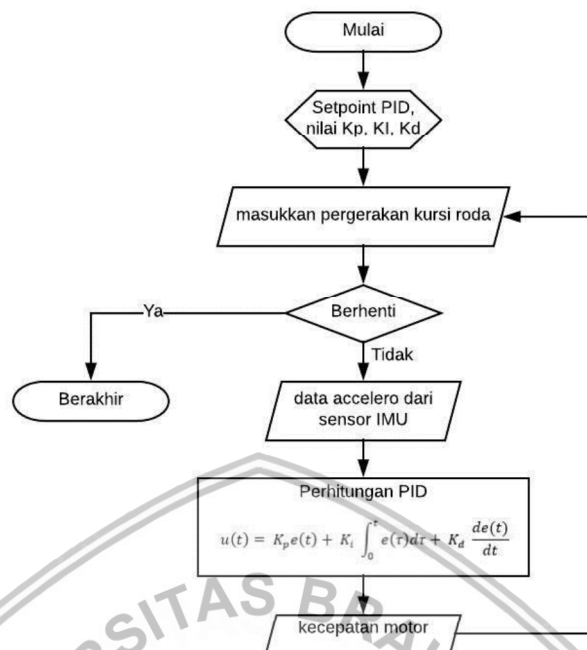
Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]				-

Gambar 5.6 Register Accelerometer configuration

Pada bit 0 hingga bit 2 tidak akan digunakan maka akan bernilai 0 (nol), agar bisa menggunakan *AFS_SEL 0* maka pada bit 3 dan bit 4 akan dibuat bernilai biner 0 (nol) pada masing – masing bit 3 dan bit 4. Kemudian pada bit 5 hingga bit 7 merupakan register yang berguna untuk mengatur *self test* pada masing – masing sudut x, sudut y, sudut z dari sensor *accelerometer*, karena pada saat pembuatan sistem ini tidak dibutuhkan *self test* maka pada bit 5 hingga 7 akan diberi nilai 0 (nol). Seluruh bit pada register konfigurasi *Accelerometer* telah ditentukan nilai yang akan dimasukkan dari bit 0 hingga bit 7. Sehingga nilai yang akan dimasukkan pada register konfigurasi *accelerometer* dalam biner adalah 0b00000000 jika diubah dalam heksa menjadi 0x00.

5.1.4 Perancangan Perangkat Lunak Kontrol PID

Perancangan kontrol PID berisi instruksi-instruksi yang akan dikerjakan oleh sistem secara otomatis. Pada Gambar 5.7 akan menjelaskan mengenai diagram alir proses PID.



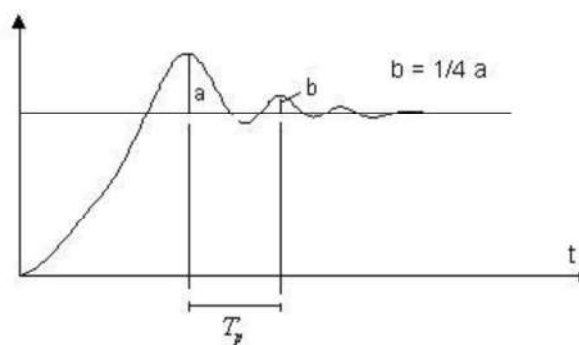
Gambar 5.7 Diagram alir proses PID

Berdasarkan Gambar 5.7 proses PID dimulai dari melakukan pemberian nilai setpoint, nilai Kp, Ki, Kd. Pengiriman masukan yang didapat dari sensor imu berupa posisi pada sumbu x, posisi pada sumbu y dan posisi pada sumbu z yang telah diubah menjadi vektor. Masukan yang didapat dari sensor akan menentukan nilai getaran untuk diolah menggunakan PID. Persamaan (5.1) Berikut ini adalah perhitungan untuk menentukan koreksi error.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (5.1)$$

Setelah nilai output didapat, motor elektrik akan bergerak sesuai dengan besaran nilai output. Lalu mengecek apakah sistem berhenti atau tidak. Jika tidak, maka sensor akan membaca kembali apakah output sudah menghasilkan nilai yang benar.

Penentuan nilai kp, ki, kd akan menggunakan metode *quarter decay Ziegler-nichols* alasan penggunaan metode ini dikarenakan sifat tumbukan yang terjadi akan menyerap energy kinetic sehingga getaran akan menghilang sesaat setelah terjadi tumbukan. Dikarena kan hal tersebut peneliti memutuskan untuk menggunakan metode quarter decay yang akan menghasilkan kurva respon seperti pada gambar 5.8



Gambar 5.8 kurva respon dari metode quarter decay

Setelah mendapatkan hasil seperti kurva respon tersebut maka akan didapatkan nilai K_u dan T_u , yang selanjutnya dapat dihitung menggunakan penalaan parameter PID seperti pada tabel 5.2

Tabel 5.2 Konversi Parameter Tuning Ziegler-Nichols

Tipe Kontroler	K_p	T_i	T_d
P	$0.5 \cdot K_u$	∞	0
PI	$0.45 \cdot K_u$	$0.5 \cdot P_u$	0
PID	$0.6 \cdot K_u$	$0.5 \cdot P_u$	$0.125 \cdot P_u$

Sumber: (Gunterus, 1994)

5.1.5 Perancangan Keluaran motor DC

Hasil keluaran dari sistem ini adalah kecepatan motor penggerak dari kursi roda yang akan melambat ketika sensor mendeteksi kondisi jalan yang tidak rata. Jika jalan terdeteksi tidak rata, maka kecepatan motor dc akan melambat. Sedangkan jika kursi roda melawati jalan yang rata seperti kondisi didalam ruangan maka kecepatan dari kursi roda akan ditingkatkan. Kondisi menaikkan kecepatan atau menurunkan kecepatan akan dijelaskan pada Tabel 5.3

Tabel 5.3 perancangan keluaran sistem motor dc

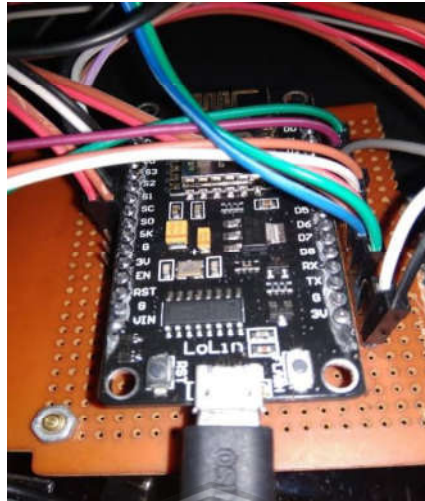
No	Kondisi jalan	Output pada motor dc
1	Rata	Menaikkan kecepatan hingga batas pwm 450 dari 1024
2	Tidak rata	Menurunkan kecepatan hingga meredam guncangan

5.2 Implementasi Sistem

Implementasi sistem kontrol kecepatan putaran motor smart wheelchair terbagi menjadi implementasi perangkat keras dan implementasi perangkat lunak. Implementasi perangkat lunak terbagi menjadi implementasi PID dan implementasi keluaran motor brushless. Berikut ini tahapan implementasi sistem.

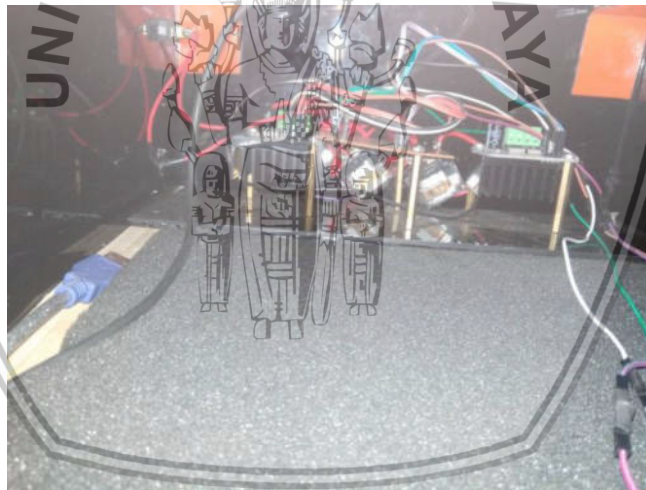
5.2.1 Implementasi Perangkat Keras

Implementasi perangkat keras sistem kontrol kecepatan putaran motor smart wheelchair adalah dengan memasang seluruh komponen perangkat keras yang dibutuhkan sistem. Komponen perangkat keras yang dibutuhkan sistem yaitu MPU 6050 dan Node MCU. Pada Gambar 5.9 berikut adalah peletakan controller NodeMCU pada bagian kursi roda.



Gambar 5.9 Peletakan kontroller

Setelah diproses sistem mengirim data kontrol ke motor penggerak kursi roda beserta nilai output ke motor penggerak kursi roda. Sistem kontrol kecepatan putaran motor smart wheelchair dibungkus dengan kotak agar lebih rapi dan menjaga agar tidak terjadi hubungan arus pendek dengan supply daya motor penggerak.



Gambar 5.10 Peletakan sistem

Pada Gambar 5.10 merupakan peletakan seluruh komponen sistem kendali kecepatan putaran motor smart wheelchair pada kursi roda. Seluruh komponen sistem diletakkan dalam tempat khusus dibawah tempat duduk kursi roda.

5.2.2 Implementasi Penempatan sensor

Pada sistem ini sensor diletakkan dibawah kursi roda, berdekatan dengan sumbu roda depan dan kaki pengguna kursi roda sesuai dengan syarat ISO-2631-1. Sensor berapa tepat berada ditengah, sejajar dengan roda kiri dan kanan. Seperti pada gambar 5.11.



Gambar 5.11 Penempatan sensor

Sensor tersebut dihubungkan pada NodeMCU untuk memproses perintah kontrol gerak dari kursi roda serta memproses kontrol kecepatan dengan PID.

5.2.3 Implementasi Pengambilan Data dari Sensor IMU

Penerapan pengambilan data dari sensor imu dilakukan pada perangkat lunak di kode program dalam node MCU. Pengambilan data dari sensor diperlukan sebagai sumber masukan sistem yang akan diolah menggunakan metode PID. Berikut ini adalah kode program untuk pengambilan data dari sensor IMU.

Tabel 5.4 Kode program pengambilan data sensor IMU

No	Kode Program
1	// MPU6050 Slave Device Address
2	const uint8_t MPU6050SlaveAddress = 0x68;
3	// Select SDA and SCL pins for I2C communication
4	////sda= d2,d3 scl=d1,d4
5	const uint8_t scl = D1;
6	const uint8_t sda = D2;
7	// sensitivity scale factor respective to full scale
8	setting provided in datasheet
9	const uint16_t AccelScaleFactor = 16384;
10	const uint16_t GyroScaleFactor = 131;
11	// MPU6050 few configuration register addresses
12	const uint8_t MPU6050_REGISTER_SMPLRT_DIV = 0x19;
13	const uint8_t MPU6050_REGISTER_USER_CTRL = 0x6A;
14	const uint8_t MPU6050_REGISTER_PWR_MGMT_1 = 0x6B;
15	const uint8_t MPU6050_REGISTER_PWR_MGMT_2 = 0x6C;
16	const uint8_t MPU6050_REGISTER_CONFIG = 0x1A;
17	const uint8_t MPU6050_REGISTER_GYRO_CONFIG = 0x1B;
18	const uint8_t MPU6050_REGISTER_ACCEL_CONFIG = 0x1C;
19	const uint8_t MPU6050_REGISTER_FIFO_EN = 0x23;
20	const uint8_t MPU6050_REGISTER_INT_ENABLE = 0x38;
21	const uint8_t MPU6050_REGISTER_ACCEL_XOUT_H = 0x3B;
22	const uint8_t MPU6050_REGISTER_SIGNAL_PATH_RESET = 0x68;
23	int16_t AccelX, AccelY, AccelZ, Temperature, GyroX, GyroY, GyroZ;

24	float acc_total_vector;
25	void readIMU(){
26	double Ax, Ay, Az, T, Gx, Gy, Gz;
27	Read_RawValue(MPU6050SlaveAddress, MPU6050_REGISTER_ACCEL_XOUT_H);
28	//divide each with their sensitivity scale factor
29	Ax = (double)AccelX/AccelScaleFactor;
30	Ay = (double)AccelY/AccelScaleFactor;
31	Az = (double)AccelZ/AccelScaleFactor;
32	acc_total_vector = sqrt((Ax*Ax)+(Ay*Ay)+(Az*Az));
33	Input = acc_total_vector; }
34	void I2C_Write(uint8_t deviceAddress, uint8_t regAddress, uint8_t data){
35	Wire.beginTransmission(deviceAddress);
36	Wire.write(regAddress);
37	Wire.write(data);
38	Wire.endTransmission(); }
39	// read all 14 register
40	void Read_RawValue(uint8_t deviceAddress, uint8_t regAddress){
41	Wire.beginTransmission(deviceAddress);
42	Wire.write(regAddress);
43	Wire.endTransmission();
44	Wire.requestFrom(deviceAddress, (uint8_t)14);
45	AccelX = (((int16_t)Wire.read()<<8) Wire.read());
46	AccelY = (((int16_t)Wire.read()<<8) Wire.read());
47	AccelZ = (((int16_t)Wire.read()<<8) Wire.read());
48	Temperature = (((int16_t)Wire.read()<<8) Wire.read());
49	GyroX = (((int16_t)Wire.read()<<8) Wire.read());
50	GyroY = (((int16_t)Wire.read()<<8) Wire.read());
51	GyroZ = (((int16_t)Wire.read()<<8) Wire.read()); }
52	//configure MPU6050
53	void MPU6050_Init(){
54	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_SMPLRT_DIV, 0x07);
55	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_PWR_MGMT_1, 0x00);
56	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_PWR_MGMT_2, 0x01);
57	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_CONFIG, 0x00);
58	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_GYRO_CONFIG, 0x00); //set +/-250 degree/second full scale

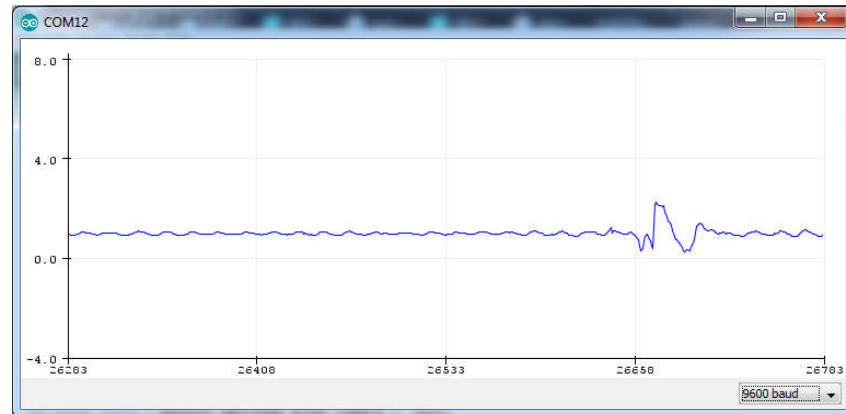
59	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_ACCEL_CONFIG, 0x00); // set +/- 2g full scale
60	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_FIFO_EN, 0x00);
61	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_INT_ENABLE, 0x01);
62	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_SIGNAL_PATH_RESET, 0x00);
63	I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_USER_CTRL, 0x00); }

Berdasarkan Tabel 5.4 kode program pada baris 2 merupakan kode program yang digunakan untuk untuk memanggil alamat I2C dari sensor IMU. Selanjutnya pada kode program baris 5 dan 6 digunakan untuk deklarasi pin yang akan digunakan untuk SDA dan SCL. Kode program pada baris 9 digunakan untuk kalibrasi nilai menggunakan 2g, sehingga nilai sensor yang didapat dibagi dengan 16384. Kode program pada baris 21-22 digunakan untuk mengakses register dari sensor IMU yang akan digunakan. Kode program pada baris 27 – 33 digunakan untuk mengambil data sensor hingga mengubah nilai sensor tersebut menjadi nilai vector dari percepatan gravitasi. Kode program pada baris 34-38 digunakan untuk memulai transmisi ke sensor IMU. Kode program pada baris 39-47 digunakan untuk mengakses register sensor, serta menerima data dari sensor dengan cara dikirimkan setiap 8 bit. Nilai sensor yang dikirim, akan disimpan untuk nantinya diolah. Pada kode program baris 52-63 digunakan digunakan untuk melakukan setting yang ada pada register yang berfungsi untuk menghidupkan buffer, menghidupkan sensor-sensor yang tersedia serta memilih nilai skala yang akan digunakan pada sensor tersebut.

5.2.4 Implementasi PID

Penerapan PID pada sistem kontrol kecepatan putaran motor smart wheelchair dilakukan pada perangkat lunak berupa kode program melalui node MCU selaku controller unit. Kode program yang dibuat dengan penerapan metode PID digunakan untuk mengatur kecepatan motor. Implementasi sistem ini dibantu dengan library PID, library ini digunakan untuk menghitung nilai output berdasarkan nilai sensor yang diterima.

Untuk menentukan nilai PID yang digunakan dengan metode Quarter Decay maka diperlukan mencari nilai ultimate gain K_u dan ultimate periode T_u . Nilai tersebut didapatkan ketika gelombang kedua sebesar $\frac{1}{4}$ dari gelombang kedua. Pada saat gelombang telah menunjukkan hal tersebut maka nilai ultimate gain didapatkan, dan nilai ultimate periode ditemukan. Berikut merupakan contoh gelombang seperti pada Gambar 5.9.



Gambar 5.12 Respon sistem ketika nilai Ku sebesar 5

Dari Gambar 5.12 tersebut ditentukan nilai Ku sebesar 5 dan Tu sebesar 0,056 nilai tersebut lalu dihitung berdasarkan nilai penalaan parameter PID seperti pada Tabel 5.5.

Tabel 5.5 Tabel Konversi Parameter Tuning Ziegler-Nichols

Tipe Kontroler	K_p	T_i	T_d
P	$0.5 \cdot K_u$	∞	0
PI	$0.45 \cdot K_u$	$0.5 \cdot T_u$	0
PID	$0.6 \cdot K_u$	$0.5 \cdot T_u$	$0.125 \cdot T_u$

Sumber: (Gunterus, 1994)

Berdasarkan tabel konversi parameter tuning Ziegler Nichols jika dihitung nilai PID dengan memasukkan nilai Ku dan nilai Tu yang didapat dari grafik respon sistem maka didapat:

$$K_p = 0,6 \times 5 = 3$$

$$T_i = 0,5 \times 0,056 = 0,028$$

$$T_d = 0,125 \times 0,056 = 0,007$$

$$K_i = k_p / t_i = 3 / 0,028 = 107,1$$

$$K_d = k_p \times t_d = 3 \times 0,007 = 0,021$$

Dari tabel 5.6 tersebut didapatkan hasil nilai $K_p = 3$, nilai $K_i = 107,1$, $K_d = 0,021$. Berikut ini adalah kode program penerapan PID pada program dalam nodeMCU.

Tabel 5.6 Kode program kontrol PID

No	Kode Program
1	//Define Variables we'll be connecting to
2	double Setpoint, Input, Output;
3	//Specify the links and initial tuning parameters
4	double Kp=3, Ki=107,1, Kd=0,021;
5	PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
6	Setpoint = 1.18;
7	myPID.SetMode(AUTOMATIC);
8	myPID.SetOutputLimits(0,MAX_SPEED-100);
9	void loop () {

10	myPID.Compute();
11	MAX_SPEED = (450-Output);}

Berdasarkan Tabel 5.5 kode program pada baris 5 yang digunakan untuk set up PID, setpoint digunakan sebagai nilai output yang diinginkan. Kode program pada baris 7 digunakan untuk mengaktifkan PID loop. Dilanjutkan dengan kode program output minimal yang diinginkan.

5.2.5 Implementasi Keluaran Motor DC

Hasil keluaran sistem kontrol kecepatan putaran motor *smart wheelchair* berupa pengaturan kecepatan motor DC. Dimana motor akan melambat pada saat berada di jalan yang tidak merata. Berikut ini pada Tabel 5.7 adalah kode program untuk mengatur keluaran motor brushless.

Tabel 5.7 Kode program keluaran motor DC

No	Kode Program
1	if ((left_fwd_increment >= INCREMENT)
2	(left_fwd_increment == -INCREMENT && left_fwd_speed > 0)) {
3	if (left_fwd_speed < MAX_SPEED && left_fwd_increment
4	!= -INCREMENT) {
5	left_fwd_speed += left_fwd_increment;
6	//left_fwd_speed = left_fwd_speed + left_fwd_increment
7	}
8	else if (left_fwd_speed > MAX_SPEED &&
9	left_fwd_increment != -INCREMENT) {
10	left_fwd_speed -= left_fwd_increment;
11	}
12	else if (left_fwd_increment == -INCREMENT) {
13	left_fwd_speed += left_fwd_increment;
14	}
15	else if (left_fwd_speed == MAX_SPEED) {
16	left_fwd_speed = left_fwd_speed;
17	}
18	analogWrite(dir2PinA, left_fwd_speed);
19	}
20	if ((right_fwd_increment >= INCREMENT)
21	(right_fwd_increment == -INCREMENT && right_fwd_speed > 0))
22	{
23	if (right_fwd_speed < MAX_SPEED &&
24	right_fwd_increment != -INCREMENT) {
25	right_fwd_speed += right_fwd_increment;
26	}
27	else if (right_fwd_speed > MAX_SPEED &&
28	right_fwd_increment != -INCREMENT) {
29	right_fwd_speed -= right_fwd_increment;
30	}
31	else if (right_fwd_increment == -INCREMENT) {
32	right_fwd_speed += right_fwd_increment;
33	}
34	else if (right_fwd_speed == MAX_SPEED) {
35	right_fwd_speed = right_fwd_speed;
36	}
37	analogWrite(dir1PinB, right_fwd_speed);
38	}

```

21     if ((reverse_increment == INCREMENT && reverse_speed <
MAX_SPEED) || (reverse_increment == -INCREMENT &&
reverse_speed > 0)) {
22         reverse_speed += reverse_increment;

23         analogWrite(dir1PinA, reverse_speed);
24         analogWrite(dir2PinB, reverse_speed);
25     } else {
26         reverse_increment = 0;
27     }
28     if (left_fwd_speed == 0 && right_fwd_speed == 0 &&
reverse_speed == 0) {
29         if (next_command == 'G') {
30             left_fwd_increment = INCREMENT;
31             right_fwd_increment = INCREMENT;

32             next_command = '_';
33         } else if (next_command == 'B') {
34             reverse_increment = INCREMENT;

35             next_command = '_';
36         } else if (next_command == 'L') {
37             right_fwd_increment = INCREMENT;

38             next_command = '_';
39         } else if (next_command == 'R') {
40             left_fwd_increment = INCREMENT;

41             next_command = '_';
42         }
43     }
44 }

45 void forward() {
46     if (left_fwd_speed > 0 && right_fwd_speed == 0){
47         right_fwd_increment = 2 * INCREMENT;
48     } else if (left_fwd_speed == 0 && right_fwd_speed > 0) {
49         left_fwd_increment = 2 * INCREMENT;
50     } else if (reverse_speed > 0) {
51         reverse_increment = -INCREMENT;

52         next_command = 'G';
53     } else {
54         left_fwd_increment = INCREMENT;
55         right_fwd_increment = INCREMENT;
56     }
57 }

58 void right() {
59     if (reverse_speed > 0) {
60         reverse_increment = -INCREMENT;

61         next_command = 'R';
62     } else {
63         left_fwd_increment = INCREMENT;
64         right_fwd_increment = -INCREMENT;
65     }
66 }

```

59	void left() {
60	if (reverse_speed > 0) {
61	reverse_increment = -INCREMENT;
62	next_command = 'L';
63	} else {
64	left_fwd_increment = -INCREMENT;
65	right_fwd_increment = INCREMENT;
	}
66	}
66	void stop_() {
67	if (left_fwd_speed > 0 right_fwd_speed > 0) {
68	left_fwd_increment = -INCREMENT;
69	right_fwd_increment = -INCREMENT;
70	} else if (reverse_speed > 0) {
71	reverse_increment = -INCREMENT;
	}
	}
72	void reverse() {
73	if (left_fwd_speed > 0 right_fwd_speed > 0) {
74	left_fwd_increment = -INCREMENT;
75	right_fwd_increment = -INCREMENT;
76	next_command = 'B';
77	} else {
78	reverse_increment = INCREMENT;
	}
	}

Berdasarkan Tabel 5.7 kode program pada baris 1-40 digunakan untuk mengecek maxspeed yang telah berubah karena hasil perhitungan PID sehingga dibutuhkan pengecekan berulang sebelum nilai PWM ditambah. Sedangkan pada baris program 41 hingga 78 digunakan untuk milih control gerak kursi roda. Pada pergerakan kursi roda dilakukan dengan cara menambahkan nilai PWM sebesar 1 atau mengurangi sebesar 1 dengan cara increment/-increment. Hal ini dilakukan agar pergerakan kursi roda menjadi lebih halus. Untuk pergerakan berbelok, cara yang digunakan mengurangi salah satu PWM motor penggerak agar bisa bergerak belok kearah yang dituju. Contohnya jika kursi roda mau digerakkan berbelok ke kanan maka pwm pada motor penggerak kanan akan dikurangi sedikit demi sedikit hingga akhirnya kursi roda akan bergerak kekanan

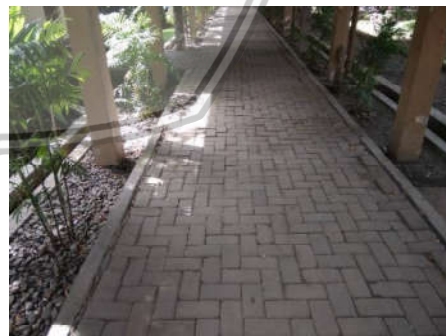
BAB 6 PENGUJIAN DAN ANALISIS SISTEM

Pada bab ini akan dibahas pengujian dan analisis hasil penelitian sistem kontrol kecepatan putaran motor *smart wheelchair* menggunakan metode PID. Pengujian dan analisis sistem ini dilakukan setelah melalui proses perancangan dan implementasi sistem. Pengujian dilakukan ditempat-tempat yang telah ditentukan seperti yang terdapat pada tabel 6.1.

Tabel 6.1 Tabel Lokasi Percobaan

Percobaan ke	Lokasi
Pertama	Di depan pintu masuk basement gedung f Fakultas Ilmu Komputer Universitas Brawijaya
Kedua	Di antara gedung f dengan gedung h
Ketiga	Lapangan parkir Filkom
Keempat	Jalan utama Universitas Brawijaya
Kelima	Koridor belakang gedung patologi
Keenam	Lapangan merah
Ketujuh	Koridor di depan gedung c
Kedelapan	Koridor di depan gedung h
Kesembilan	Lapangan parkir Fakultas Kedokteran Gigi Brawijaya
Kesepuluh	Parkiran Patologi

Pengujian pertama dilakukan dengan membandingkan hasil sebelum menggunakan PID dengan sesudah menggunakan PID untuk mengetahui apakah sistem memberikan hasil yang signifikan. Pengujian kedua dilakukan dengan apakah sistem yang digunakan mampu meredam getaran yang diterima.



Gambar 6. 1 Contoh salah satu lokasi yang digunakan pada pengujian

Pengujian ketiga dilakukan dengan standar ISO-2631-1 untuk mengetahui apakah sistem telah berfungsi dan serta sesuai standar ISO yang berlaku pada jalan seperti pada Gambar 6.1.

6.1 Pengujian dan analisis Putaran motor DC Pada Jalan Rata dan Pada Jalan Concrete Blok

6.1.1 Tujuan

Tujuan dari pengujian ini adalah mengetahui apakah PWM pada motor penggerak akan berkurang jika menghadapi jalan tidak rata dan akan berada di nilai pwm 450 jika berada pada jalan rata.

6.1.2 Alat

Alat yang digunakan untuk pengujian putaran motor dc berupa:

1. Sensor IMU
2. NodeMCU
3. Komputer
4. Kursi roda

6.1.3 Metode

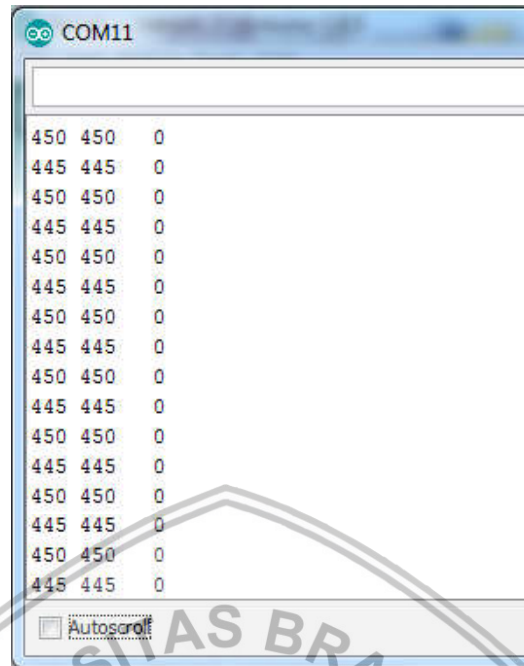
Metode yang dilakukan untuk pengujian akurasi sensitivitas sensor IMU adalah sebagai berikut:

1. Melakukan perakitan komponen berupa sensor dan nodeMCU
2. Menghubungkan nodeMCU dengan Komputer dan sumber daya
3. Menjalankan program serta melakukan tuning PID
4. Memberikan input berupa perintah jalan dan menjalankan kursi roda di jalanan *concrete block*
5. Mengamati dan mengambil data keluaran sistem
6. Menentukan tingkat keberhasilan sistem dengan persamaan 6.1 berikut:

$$\text{Tingkat keberhasilan} = \frac{\text{jumlah berhasil}}{\text{Banyak percobaan}} \times 100\% \quad (6.1)$$
7. Menganalisis hasil dan mengambil kesimpulan

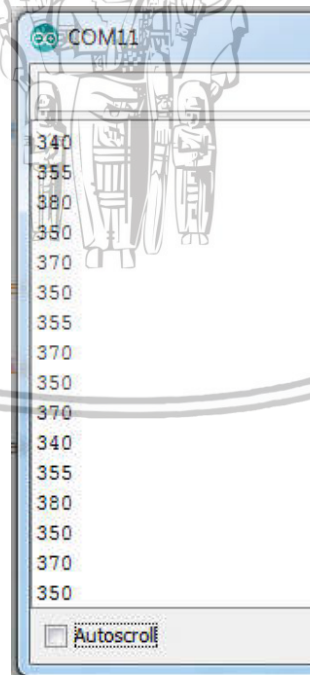
6.1.4 Hasil

Hasil yang didapat berdasarkan pengujian jalan rata didalam ruangan dengan lantai keramik dan pada jalan concrete blok. Berikut merupakan tampilan nilai pwm pada roda kiri dan kanan saat kondisi maju, serta pwm untuk mundur pada serial monitor saat jalan rata pada Gambar 6.2



Gambar 6.2 Nilai Pwm Jalan Rata

Pada Gambar 6.2 merupakan hasil pwm yang didapat dari jalan pada lantai keramik. Setelah didapat hasil pengujian dari *serial monitor*. Selanjutnya percobaan dilakukan pada jalan concrete blok.



Gambar 6.3 Nilai pwm jalan tidak rata

Pada Gambar 6.3 merupakan tampilan pwm pada serial monitor. Gambar tersebut merupakan hasil pwm yang didapat dari jalan pada concrete blok. Setelah didapat hasil pengujian dari *serial monitor*.

6.1.5 Analisis

Dari hasil kedua tabel tersebut terdapat sedikit perubahan pada pwm dikarenakan sistem mendeteksi adanya sambungan antar keramik. Sedangkan pada jalan tidak rata, sistem dapat menurunkan kecepatan berdasarkan kondisi jalan tidak rata.

6.2 Pengujian dan Analisis sistem kendali kecepatan untuk mengurangi getaran

6.2.1 Tujuan

Tujuan dari pengujian sistem kendali kecepatan untuk mengurangi getaran adalah untuk mengetahui apakah sistem yang dibuat dapat berjalan dengan baik dan mampu mengurangi getaran mekanis yang terjadi selama kursi roda digunakan dibandingkan sistem yang tidak menggunakan pengendali kecepatan. Data yang didapat dari sensor IMU akan digunakan untuk masukan proses PID.

6.2.2 Alat

Alat yang digunakan untuk pengujian kendali kecepatan berupa:

1. Sensor IMU
2. NodeMCU
3. Komputer
4. Kursi roda

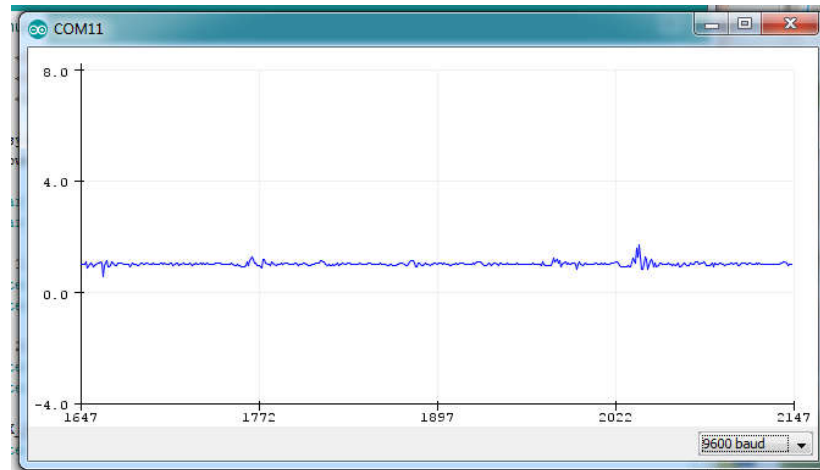
6.2.3 Metode

Metode yang dilakukan untuk pengujian akurasi sensitivitas sensor IMU adalah sebagai berikut:

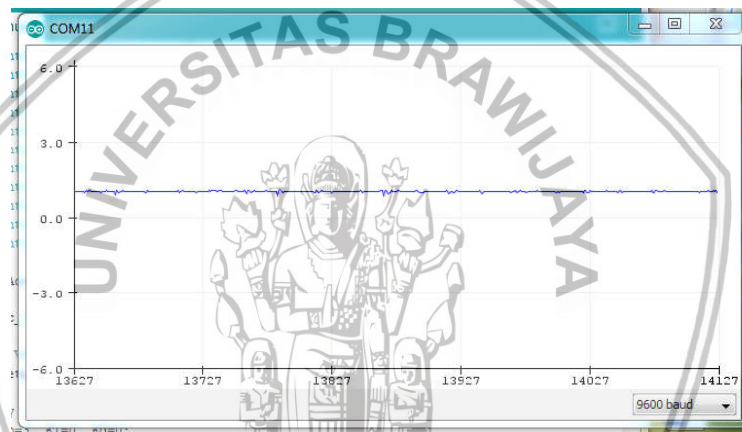
1. Melakukan perakitan komponen berupa sensor dan nodeMCU
2. Menghubungkan nodeMCU dengan Komputer dan sumber daya
3. Menjalankan program serta melakukan tuning PID
4. Memberikan input berupa perintah jalan dan menjalankan kursi roda di jalanan *concrete block*
5. Mengamati dan mengambil data keluaran sistem
6. Menentukan tingkat keberhasilan sistem dengan persamaan 6.1
7. Menganalisis hasil dan mengambil kesimpulan

6.2.4 Hasil

Pengujian dilakukan diberbagai tempat disekitaran Universitas Brawijaya yang menggunakan *concrete block*. Pengujian pun dilakukan sebanyak 10 kali ditempat yang berbeda. Berikut merupakan perbandingan antara sistem yg menggunakan PID dengan yang belum menggunakan PID ditunjukkan pada Gambar 6.4 dan Gambar 6.5

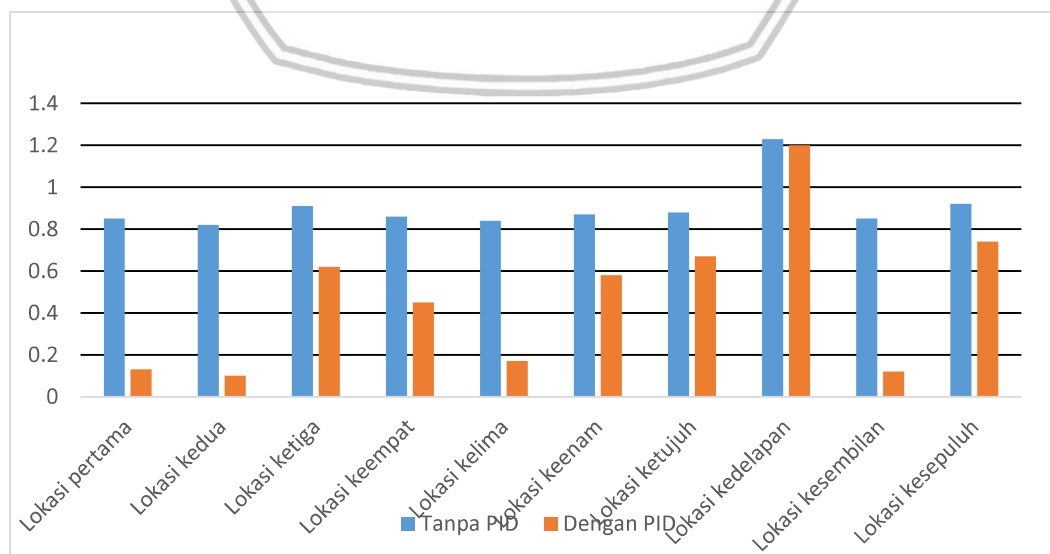


Gambar 6.4 Hasil pembacaan sensor pada sistem tanpa PID



Gambar 6.5 Hasil setelah menggunakan control kecepatan dengan PID

Hasil pengujian performa sistem dalam meredam getaran dapat dilihat pada Gambar 6. 6 berikut.



Gambar 6.6 Hasil Pengujian Perbandingan Getaran

6.2.5 Analisis

Pengujian sistem kendali kecepatan dilakukan sebanyak sepuluh kali. Dari sepuluh pengujian, didapat sembilan pengujian dengan nilai persentase diatas 20% dan satu pengujian dengan nilai persentase kurang dari 5%. Terdapat hasil tes persentase meredam dengan buruk dikarenakan kontur jalan yang sangat rusak. Dengan hasil tersebut, sistem dapat dianggap bekerja dengan baik. Peredaman getaran mekanis tidak dapat mencapai angka maksimal dikarenakan bukan hanya perbedaan elevasi dari penempatan *concrete block* namun juga berpengaruh pada sudut penempatan serta golongan *concrete block* yang digunakan.

6.3 Pengujian dan Analisis sistem kendali kecepatan untuk mengurangi getaran dengan standar ISO-2631-1

6.3.1 Tujuan

Tujuan dari pengujian sistem kendali kecepatan untuk mengurangi getaran dengan standar ISO-2631-1 adalah untuk mengetahui apakah sistem sudah lolos ISO-2631-1 atau belum. Hasil *output* yang didapat setelah proses ini adalah kecepatan motor yang melambat atau meningkat. Jika sistem dapat melakukan kontrol dengan baik, maka tujuan dari pengujian kontrol PID telah tercapai.

6.3.2 Alat

Alat yang digunakan untuk pengujian kontrol PID berupa:

1. Sensor IMU
2. NodeMCU
3. Komputer
4. Kursi roda

6.3.3 Metode

Metode yang dilakukan untuk melaksanakan pengujian dengan standar ISO-2631-1 adalah sebagai berikut:

1. Melakukan perakitan komponen berupa sensor IMU dan node MCU
2. Menghubungkan nodeMCU dengan Komputer dan sumber daya
3. Menjalankan program kontrol PID
4. Memberikan input berupa kondisi jalan yang berbeda pada kursi roda
5. Mengamati dan mengambil data keluaran sistem
6. Menentukan tingkat keberhasilan sistem dengan persamaan (6.1).
7. Menganalisis hasil dan mengambil kesimpulan

6.3.4 Hasil

Berdasarkan pada ISO-2631-1 sistem dianggap nyaman jika getaran bernilai dibawah $0,8 \text{ m/s}^2$. Selain itu tingkat peredaman getaran dibagi menjadi beberapa bagian sesuai dengan Tabel 6.2 berikut.

Tabel 6.2 level kenyamanan tubuh pengguna ketika terpapar getaran mekanis berdasarkan ISO 2631-1 (1997)

Nilai percepatan (m/s ²)	Skala kenyamanan
Kurang dari 0.315	Nyaman
0.315 – 0,5	Sedikit kurang nyaman
0,5 – 0,8	Agak tidak nyaman
0,8 – 1,25	Tidak nyaman
1,25 – 2	Sangat tidak nyaman
Lebih dari 2	Amat sangat tidak nyaman

Sumber: (International Organization for Standardization, 1997)

Berikut adalah hasil pengujian kontrol PID yang dilakukan pada beberapa kondisi jalan akan ditunjukkan pada Tabel 6.3.

Tabel 6.3 Hasil Pengujian kontrol PID dengan berbagai kondisi jalan

No	Lokasi	Skala kenyamanan					
		Nyaman	Sedikit kurang nyaman	Agak tidak nyaman	Tidak nyaman	Sangat tidak nyaman	Amat sangat tidak nyaman
1	Pertama	√	×	×	×	×	×
2	Kedua	√	×	×	×	×	×
3	Ketiga	√	×	×	×	×	×
4	Keempat	√	×	×	×	×	×
5	Kelima	√	×	×	×	×	×
6	Keenam	×	√	×	×	×	×
7	Ketujuh	√	×	×	×	×	×
8	Kedelapan	×	×	×	√	×	×
9	Kesembilan	√	×	×	×	×	×
10	Kesepuluh	√	×	×	×	×	×
Keberhasilan		80%	10%	0%	10%	0%	0%

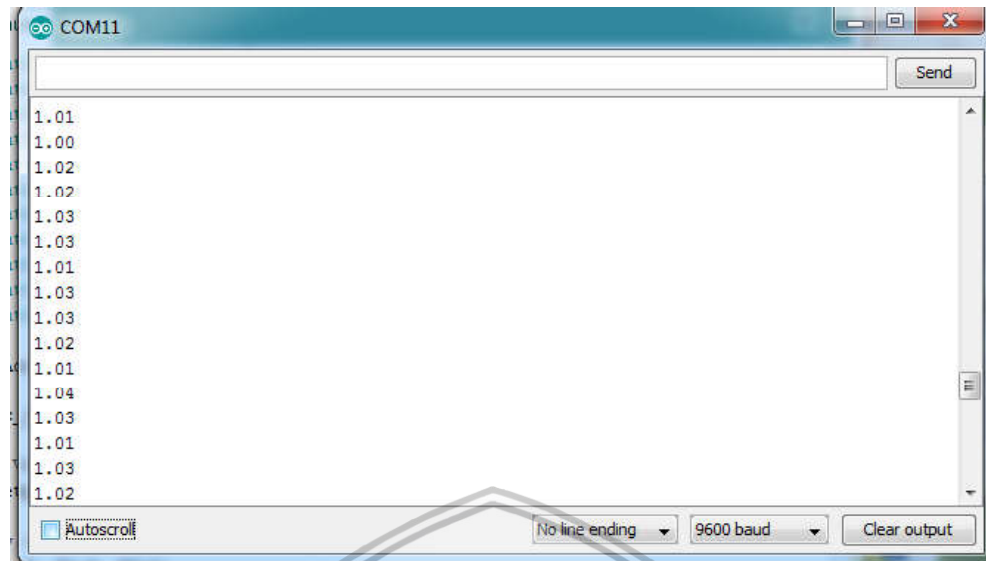
Keterangan:

1. √ : Terdeteksi
2. × : Tidak terdeteksi

Dari Tabel 6.3 didapat rata-rata nilai Persentase keberhasilan sebesar:

$$\text{Rata – rata keberhasilan} = \frac{8}{10} \times 100\% = 80\%$$

Dari Tabel 6.2 didapat hasil 80% nyaman sehingga sistem sudah dapat dikatakan mampu meredam getaran mekanis dari jalan *concrete block*. Berikut adalah hasil pada jalan tidak rata bernilai seperti pada Gambar 6.7 pada gambar tersebut memiliki nilai stabil 1,04 yang menandakan jalan rata, sedangkan jika nilai berbeda dari itu maka jalan bergetar. Semakin besar nilai, getaran semakin besar



Gambar 6.7 Hasil nilai getaran

6.3.5 Analisis

Berdasarkan Tabel 6.3 adalah hasil dari pengujian sistem dengan PID untuk meredam getaran dan syarat nilai baku dari ISO-2631-1. Dari hasil tersebut 80 % getaran mampu diredam hingga mampu masuk dalam klasifikasi nyaman. Sedangkan yang lain terpengaruh pada faktor sudut yang dibentuk dari *concrete block* menjadi polisi tidur maupun cekungan, sehingga memberikan getaran yang lebih kuat dan tidak mampu diatasi

BAB 7 PENUTUP

Pada Bab penutup akan dibahas kesimpulan yang didapat dari proses penelitian kendali kecepatan untuk meredam getaran pada jalan bergelombang dengan metode pid yang telah dilakukan sebelumnya. Selain itu, penutup juga membahas saran untuk pengembangan penelitian yang telah dilakukan.

7.1 Kesimpulan

Berdasarkan proses penelitian yang sudah dilakukan, rumusan masalah yang telah dibuat sebelumnya dan dari hasil penelitian yang sudah ada, maka dapat disimpulkan bahwa:

1. Sistem secara keseluruhan mendapat hasil yang cukup baik dengan hasil perbandingan sistem yang menggunakan kontrol PID 9 kali lebih mampu meredam getaran dibandingkan dengan sistem yang tidak menggunakan kontrol PID
2. Kontrol PID untuk menentukan kecepatan motor dapat digunakan untuk mengurangi getaran mekanis. Pada penelitian ini kecepatan motor diperlambat untuk mengurangi momentum tabrakan antara roda dengan jalan yang tidak rata sehingga dapat mengurangi getaran dari efek tabrakan tersebut. Sedangkan untuk mengetahui berapa tekanan yang terjadi pada saat tabrakan terjadi adalah sensor IMU, nilai yg didapat dari sensor dijadikan sebagai input dari PID
3. Hasil pengujian sistem kendali kecepatan putaran motor smart wheelchair untuk meredam getaran pada jalan bergelombang dengan metode PID dan pengujian berdasarkan standar ISO-2631-1 sebanyak 80% terbukti nyaman.

7.2 Saran

Saran yang diberikan untuk sistem ini guna penelitian yang lebih lanjut kedepannya berupa:

1. Menggunakan sensor accelerometer atau sensor IMU dengan kualitas yang lebih baik serta tahan terhadap guncangan
2. Menggunakan sensor accelerometer atau sensor IMU lebih dari satu, dengan penempatan sensor pada roda, dibawah tempat duduk dan dibelakang sandaran punggung. Dengan begitu sensor dapat mendeteksi bagian mana dari tubuh yang terpapar getaran dari jalan

Daftar Pustaka

- Arduino, 2018. *Arduino IDE*. [Online] Tersedia di:
<https://www.arduino.cc/en/Main/Software> [Diakses pada 20 September 2018].
- Arduino, 2018. *MPU-6050 Accelerometer + Gyro*. [Online] Tersedia di:
<http://playground.arduino.cc/Main/MPU-6050> [Diakses pada 20 September 2018].
- Barnett, C., 2007. *Embedded C Programming and the Atmel AVR 2nd edition*. Canada: Nelson Education.
- Clinton, A., Syaury, D. & Utaminingrum, F., 2018. Sistem Monitoring RPM Roda Smart Wheelchair Pada Halaman Web Berbasis Ajax Menggunakan Sensor Optocoupler. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*.
- Dan Chen, D. E. S., 2002. PI/PID Controller Design Based on Direct Synthesis and. *IEEE*.
- Gunterus, F., 1994. *Falsafah Dasar: Sistem Pengendalian Proses*. Jakarta: PT. Elex Media Komputindo.
- Hilberink, S. R., 2006. *Principles of Electric Wheelchair Hockey*. Aylesbury, UK: International Wheelchair & Amputee Sports Federation (IWAS).
- International Organization for Standardization, 1997. *ISO 2631-1*. [Online] Tersedia di:
<https://www.iso.org/standard/7612.html> [Diakses pada 20 September 2018].
- Mazidi, M. A., 2011. *The Microcontroller and Embedded System: Using Assembly and C*. New Jersey: Pearson Education.
- Noriaki Hirose, K. S. Y. T., 2012. Suppression of Roll Vibration for Personal Mobility. *IEEE*.
- Ogata, K., 2010. *Modern Control Engineering*. New Jersey: Pearson Education, Inc.
- Rosidawati, Jubaedi, A. & Batubara, I., 2008. *Mengenal Usia Lanjut*. Jakarta: s.n.
- Wang, T., Kaneko, J. & Kojima, K., 2017. Study on Relevance Between Electric Wheelchair. *IEEE*.
- Zerynth, 2016. *NodeMCU v3*. [Online] Tersedia di:
<https://docs.zerynth.com/latest/official/board.zerynth.nodemcu3/docs/index.html> [Diakses pada 20 September 2018].